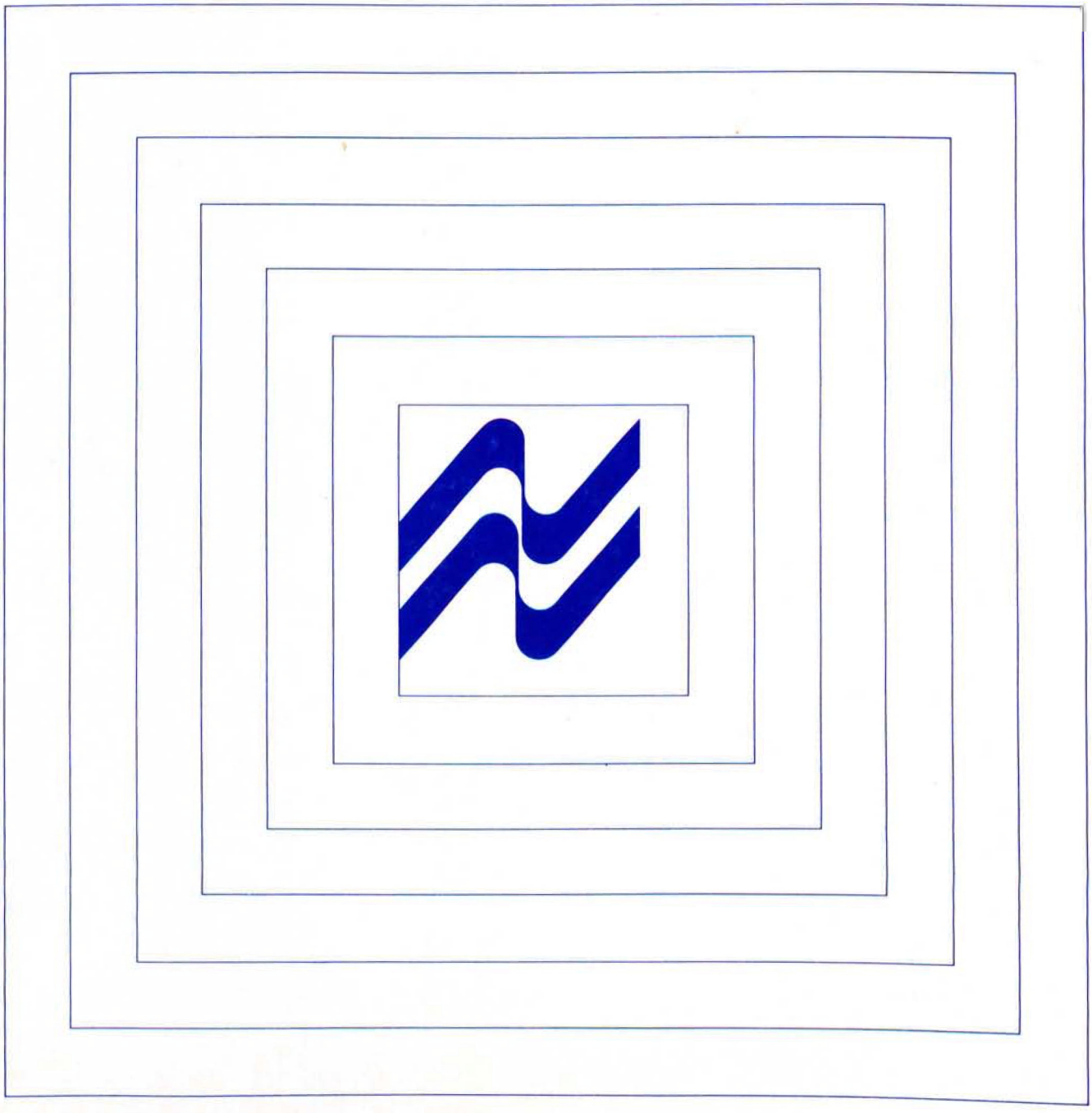
Users Manual

National Semiconductor

SC/MP Low Cost Development System



PREFACE

This manual comprises user information pertaining to the SC/MP Low Cost Development System (LCDS). The following coverage is provided:

- A general description of the LCDS
- Installation and operational instructions
- Programming examples keyed to the routines associated with the operational procedures
- Functional descriptions that interrelate the resident firmware program, the keyboard and display panel, and the logic and buffering circuits used for transferring control back and forth between the resident firmware program and a user-entered application program
- Servicing guidelines

For purposes of user applications system development, the user should have a working knowledge of computer programming, digital circuit logic, and integrated circuits.

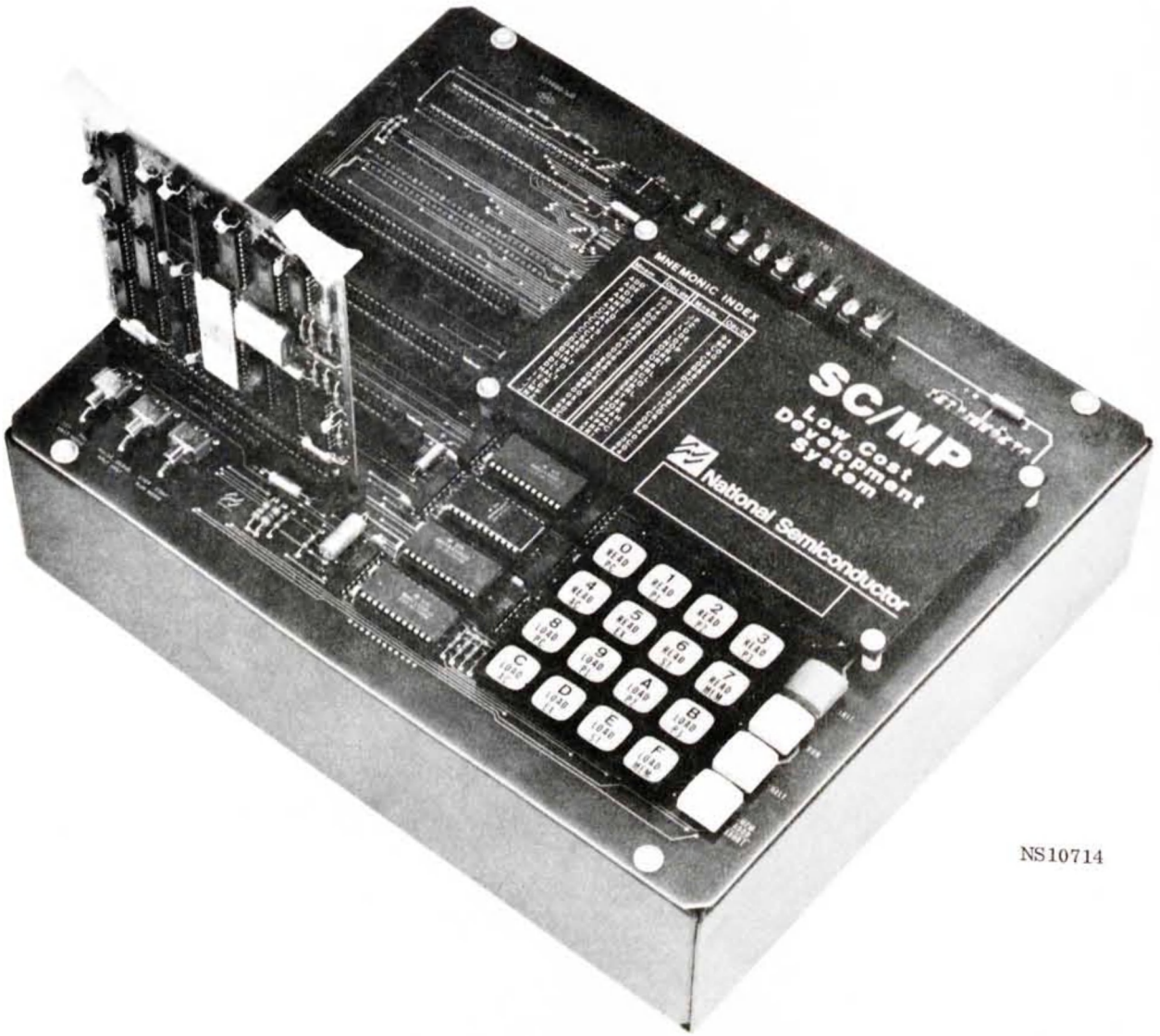
The material in this manual is subject to change without notice. Circuit details and other data presented in the engineering documentation supplied with the SC/MP LCDS take precedence over the information presented in this manual.

Copies of the following publications that are relevant to the SC/MP LCDS may be obtained from the local National Semiconductor sales office.

- SC/MP Programming and Assembler Manual (NSC Order No. ISP-8S/994Y)
- SC/MP Technical Description (Publication No. 4200079)
- Data Sheet — ISP-8A/500D Single-Chip 8-bit Microprocessor (SC/MP)
- Data Sheet — ISP-8C/100 CPU Application Card
- Data Sheet — ISP-8C/004P 4K-by-8 PROM Application Card & ISP-8C/004B 4K-by-8 ROM/PROM Socket Application Card
- Data Sheet — ISP-8C/002 2K-by-8 Read/Write Memory Application Card

TABLE OF CONTENTS (Continued)

LIST OF TABLES



NS10714

- 20-MILLIAMPERE TELETYPE INTERFACE — (continued)
  - Initiate execution of user-generated applications program at any memory address
  - Obtain applications program load module by punching selected memory range to paper tape
  - Load LCDS-generated paper-tape load module into memory
  - Load IMP-16 or FORTRAN Cross Assembler-generated paper-tape load module into memory

1.2 LEADING PARTICULARS

LCDS electrical and physical characteristics are listed in table 1-1.

Table 1-1. Leading Particulars

Feature	Description
Data Word Length	8 bits
Addressing	16 bits
Instruction Set	46 Instructions
Arithmetic	Parallel, binary, fixed point, twos complement and parallel 2-digit BCD
Memory	8-bit bytes of semiconductor memory expandable to a maximum of 65,536 bytes
Addressing Modes	PC-Relative Immediate Indexed Auto-Indexed
Typical Operating Speed	2.0 microseconds/microcycle
Input/Output and Control	8-bit buffered data bus 16-bit buffered address bus 1 serial data-input port 1 serial data-output port 3 general-purpose control flag outputs 1 general-purpose sense input 1 general-purpose sense/interrupt input Bus-request output and bus-enable input (permit use of allocation logic for Direct Memory Access and multi-processor applications)
Input Power (at 25°C)	+5 VDC (current specifications are provided -12 VDC in chapter 2)
Humidity	Maximum of 90-percent relative humidity without condensation
Dimensions of Chassis	4 inches (10.2 centimeters) high 12 inches (30.0 centimeters) wide 10 inches (25.4 centimeters) deep

Chapter 2

INSTALLATION

2.1 DC POWER REQUIREMENTS

The LCDS is designed to operate on user-supplied +5VDC and -12VDC input power. Since the LCDS provides a plug-in interface for user-fabricated SC/MP application systems, overall current requirements are determined by adding the current requirements of the LCDS chassis to the current requirements of the applications hardware with which it is interconnected. Current requirements for the LCDS chassis and for SC/MP applications cards are listed in table 2-1; current requirements for other applications hardware should be specified in the manufacturer's literature.

2.2 PREPARATION FOR USE

To prepare the LCDS for initial operation, refer to figure 2-1 and proceed as follows.

CAUTION

To prevent damage to equipment, always turn off power to LCDS before connecting or disconnecting SC/MP Application Cards or other SC/MP applications hardware.

1. Insert the SC/MP CPU Application Card into one of the four prewired sockets (J1 through J4) with the component side of the card facing the front of the LCDS chassis.
2. Verify that a jumper is installed between pins 5 and 12 on socket J8 and that the remaining pins on the socket are unterminated. (The functions of the jumper options available at socket J8 are described in table 3-2 of chapter 3, Operation, and instructions for using the jumper options to effect various special-purpose configurations are provided in 3.5, Application System Interfacing.)
3. Using 16-gauge or heavier wire, connect +5VDC and -12VDC power to LCDS. If separate external power supplies are employed, both power supplies must be referenced to common GND terminal.

Table 2-1. LCDS Power Requirements

Item	Current Requirements	
	+5VDC	-12VDC
Low-Cost Development System Chassis (without SC/MP CPU Application Card installed)	1.5A	300mA
SC/MP CPU Application Card (with MM5204 in PROM socket)	600mA	150mA
SC/MP Read/Write Memory Application Card	1.45A	NA
SC/MP ROM or ROM/PROM Application Card (with eight MM5204 PROMs installed)	550mA	320mA

2.3 PANEL CHECKOUT

To verify the proper operation of the LCDS controls and indicators, perform the steps listed below. The purpose of each step is listed in table 2-2, along with a reference to the appropriate description provided in chapter 3, Operation.

1. Set HALT MODE switch to PNL. Then turn on +5VDC and -12VDC power to LCDS, and verify that digital readout displays 0001 PC.
2. Press READ P1 pushbutton, and verify that digital-readout display changes to 0000 P1.
3. Press READ P2 pushbutton, and verify that digital-readout display changes to 0000 P2.
4. Press READ P3 pushbutton, and verify that digital-readout display changes to 0000 P3.
5. Press READ AC pushbutton, and verify that digital-readout display changes to 00 A.
6. Press READ EX pushbutton, and verify that digital-readout display changes to 00 E.

NOTE

The term "XX" or "XXXX" is used to denote uncertainty of hexadecimal value displayed.

7. Press READ ST pushbutton, and verify that digital-readout display changes to 10 S.
8. Press READ MEM pushbutton, and verify that digital-readout display changes to 0000 "XX".
9. Press LOAD PC pushbutton, and verify that digital-readout display changes to ---- PC. Then, key in any 2-digit value (XX--PC), press MEM ADDR/ABORT pushbutton, and verify that digital-readout display changes to □□□□ □□.
10. Press READ PC pushbutton, and verify that digital-readout display changes to 0001 PC.
11. Press MEM ADDR/ABORT pushbutton, and verify that digital-readout display changes to ----. Then, key in value 7800, and verify that last two digits of digital-readout display C8 (contents of memory location 7800).

NOTE

If a wrong digit is accidentally keyed in when a numeric value is being entered for a LOAD Command, press the MEM ADDR/ABORT pushbutton to terminate the command (digital readout will display □□□□ □□). Then, press the LOAD pushbutton again to reselect the command, and key in the correct data value.

Address 7800 is a Read-Only memory location.

12. Press LOAD MEM pushbutton, and verify that digital-readout display changes to 7800 --. Then, key in any value except C8, and verify that digital readout display changes to □□□□ □□.
13. Press MEM ADDR/ABORT pushbutton, and verify that digital-readout display changes to ----. Then, key in value 5555, and verify that last two digits of digital readout display "XX" (contents of memory location 5555).
14. Press LOAD MEM pushbutton, and verify that digital-readout display changes to 5555 --. Then key in value 55 and verify that digital-readout display changes to 5555 55.
15. Press MEM ADDR/ABORT pushbutton, and verify that digital-readout display changes to ----. Then, key in value AAAA, and verify that last two digits of digital readout display "XX" (contents of memory location AAAA).
16. Press LOAD MEM pushbutton, and verify that digital-readout display changes to AAAA --. Then, key in value AA, and verify that digital-readout display changes to AAAA AA.

Memory Location	Data Value	Instruction		Comments
7701 7702	C4 20	LDI	020	Load AC with low-order digital-readout address.
7703	31	XPAL	P1	Exchange P1-low with AC.
7704 7705	C4 70	LDI	070	Load AC with high-order digital-readout address.
7706	35	XPAH	P1	Exchange P1-high with AC.
7707 7708	C4 77	LDI	077	Load AC with digital-readout display code for hexadecimal value A.
7709 770A	C9 00	ST	0(P1)	Display hexadecimal value A on first digit of digital readout.
770B 770C	C4 07	LDI	07	Load AC with loop-count value 07.
770D 770E	C8 64	ST	.+100	Store contents of AC at memory location "LOOP" (current PC address $+100_{10}$).
770F 7710	8F FF	DLY	0FF	Delay Instruction.
7711 7712	B8 60	DLD	.+96	Decrement memory location "LOOP" (current PC address $+96_{10}$).
7713 7714	9C FA	JNZ	.-6	Repeat delay/decrement loop until contents of memory location "LOOP" are decremented to zero.
7715	00	HLT		Halt Instruction.
7716 7717	C4 7C	LDI	07C	Load AC with digital-readout display code for hexadecimal value b.
7718 7719	C9 00	ST	0(P1)	Display hexadecimal value b on first digit of digital readout.
771A 771B	C4 07	LDI	07	Load AC with loop-count value of 07.
771C 771D	C8 55	ST	.+85	Store contents of AC at memory location "LOOP" (current PC address $+85_{10}$).
771E 771F	8F FF	DLY	0FF	Delay Instruction.
7720 7721	B8 51	DLD	.+87	Decrement memory location "LOOP" (current PC address $+81_{10}$).
7722 7723	9C FA	JNZ	.-6	Repeat delay/decrement loop until contents of memory location "LOOP" are decremented to zero.
7724 7725	90 E1	JMP	.-31	Jump back to instruction that loads AC with digital-readout display code for hexadecimal value A.

33. Press MEM ADDR/ABORT pushbutton, and key in value 7701.

34. Use READ MEM pushbutton to examine memory locations 7701 through 7725 sequentially, and verify that the program was loaded properly. If an error is detected, press MEM ADDR/ABORT pushbutton, and key in address of memory location containing erroneous data value. Then, press LOAD MEM pushbutton, and key in correct data value.

Table 2-2. Panel Checkout Description (Continued)

Step	Purpose	Reference
32-34	These steps verify that a memory-reference address can be stored via the MEM ADDR/ABORT pushbutton and that a program can be loaded via the LOAD MEM pushbutton.	3.3.3.4
35, 36, 37	These steps verify the LCDS program-execution STEP function; that is, one instruction of the program is executed each time that the RUN pushbutton is pressed; then the LCDS returns to the DEBUG Mode.	3.3.1
38	This step verifies that an applications program can be executed normally when the RUN pushbutton is pressed and the RUN MODE switch is set to CONTIN.	3.3.1
39	This step verifies that the LCDS will terminate the RUN Mode and enter the DEBUG Mode when the HALT INST switch is set to DEBUG and a Halt Instruction is executed.	3.3.1
40	This step verifies that RUN Mode operation will not be affected by execution of a Halt Instruction when the HALT INST switch is set to PULSE.	3.3.1
41	This step verifies that the LCDS will terminate the RUN Mode and enter the DEBUG Mode when the HALT pushbutton is pressed.	3.3.1

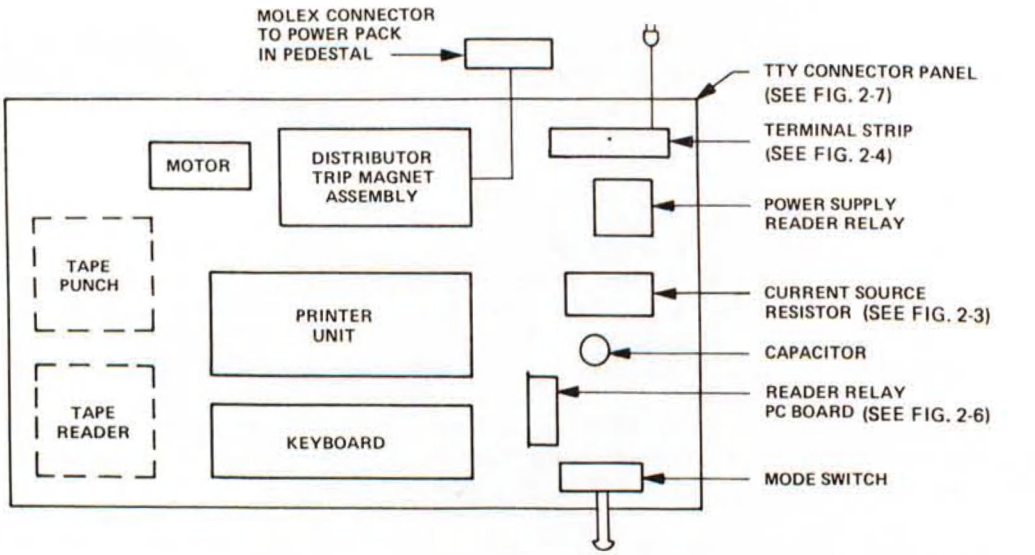


Figure 2-2. TTY Layout

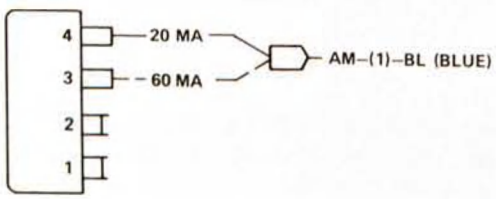


Figure 2-3. TTY Current Source Resistor

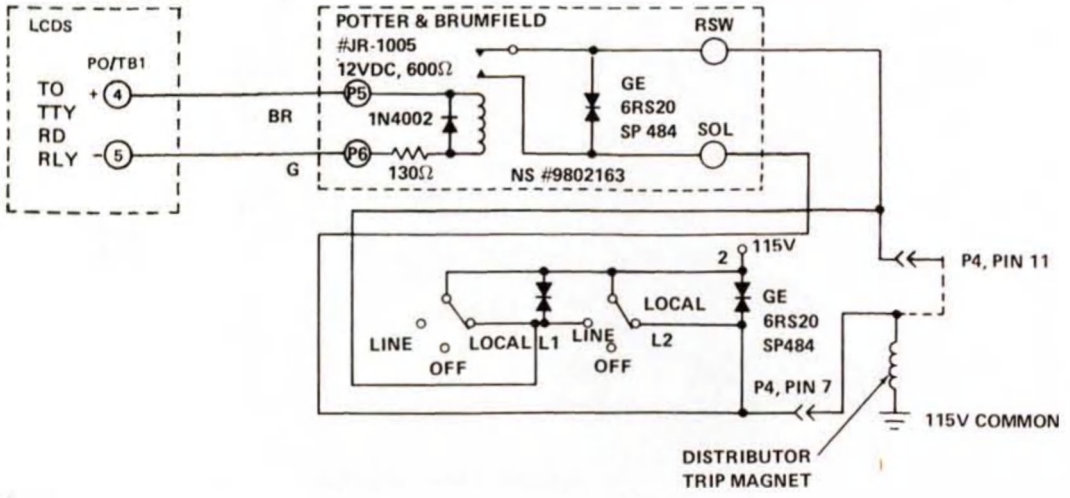
2.4.2 Paper Tape Reader Relay Option

If the paper tape reader relay option is required, the following additional steps must be performed.

NOTE

A board containing the required reader relay and associated circuits can be obtained from the National Semiconductor Corporation. The order number is IPC-16P/810R. Alternately, the user can provide the relay and circuit. Details of the relay board are shown in figure 2-6.

1. Install relay board, NS Number IPC-16P/810R or equivalent, in Paper Tape Reader drive circuit. Mounting tab with holes for mounting board is located in lower corner next to keyboard. Mount board with components facing away from keyboard (figure 2-2) utilizing two number 6 screws and lockwashers.
2. On jack 4 (male) at rear of Teletype, remove brown wire and pin from pin 11 and install at pin 7 (figure 2-7).
3. Connect wire from plug 4 (female), pin 7, at rear of Teletype to SOL terminal lug on relay board, using Molex termination at pin 7 and clip-on termination at SOL terminal lug. Then, connect second wire from SOL terminal lug to terminal L2 on Line/Local switch at front of Teletype (figure 2-6).
4. Run wire from L1 on Line/Local Switch to RSW terminal lug on relay board using clip-on termination.
5. Connect TO TTY RD RLY + output of LCDS to P5 terminal lug on relay board (figure 2-6).
6. Connect TO TTY RD RLY - output of LCDS to P6 terminal lug on relay board (figure 2-6).
7. Install two thyrectors (transient suppressors) on Line/Local Switch. Connect one between terminals L2 and 2 and the other between terminals L1 and 2 (figure 2-6).



NS10509

Figure 2-6. Reader Relay Schematic

7. On TTY, type G7701 and press RETURN key. Verify that first digit of digital readout alternately displays hexadecimal values A and b, and that A is first value displayed after RETURN key is pressed.

8. On LCDS, set HALT INST switch to DEBUG. Verify that after a few seconds digital readout blanks and TTY prints out:

   CL 7716
   -

9. On TTY, type G and press RETURN key. Verify that the following indications are obtained in the sequence listed:

   a. Hexadecimal value b is displayed on first digit of digital readout for a few seconds, then display changes to hexadecimal value A.

   b. After hexadecimal value A is displayed for a few seconds, digital readout blanks and TTY prints out:

      CL 7716
      -

10. On TTY, type A 7715, 08 and press RETURN key. Then, type H7724, and press RETURN key a second time.

11. On TTY, type G7701 and press RETURN key. Verify that the following indications are obtained in the sequence listed:

   a. Hexadecimal value A is displayed on first digit of digital readout for a few seconds, then display changes to hexadecimal value b.

   b. After hexadecimal value b is displayed for a few seconds, digital readout blanks and TTY prints out:

      CL 7724
      -

12. On TTY, type H and press RETURN key. Then type G and press RETURN key a second time. Verify that first digit of digital readout alternately displays hexadecimal values A and b, and that A is the first value displayed after the RETURN key is pressed.

13. On LCDS, set RUN MODE switch to STEP. Verify that digital readout blanks and that TTY prints out:

    CL "XXXX" (any 4-digit hexadecimal value between 7707 and 7724)
    -

14. On LCDS, press INIT switch and verify that TTY prints out:

    CL 0001
    -

15. On TTY, type G7701 and press RETURN key. Verify that TTY prints out:

    CL 7703
    -

16. On TTY, type G and press RETURN key. Verify that TTY prints out:

    CL 7704
    -

17. Turn off TTY and LCDS to erase stored program. Then turn TTY and LCDS back on, press INIT pushbutton, and verify that TTY prints out:

    CL 0001
    -

18. On TTY, type in program data value list as indicated below, then press RETURN key.

    -A7701, C4, 20, 31, C4, 70, 35, C4, 77, C9, 00, C4, 07, C8, 64, 8F, FF, B8, 60, 9C, FA, 00

Table 2-3. TTY Checkout Description

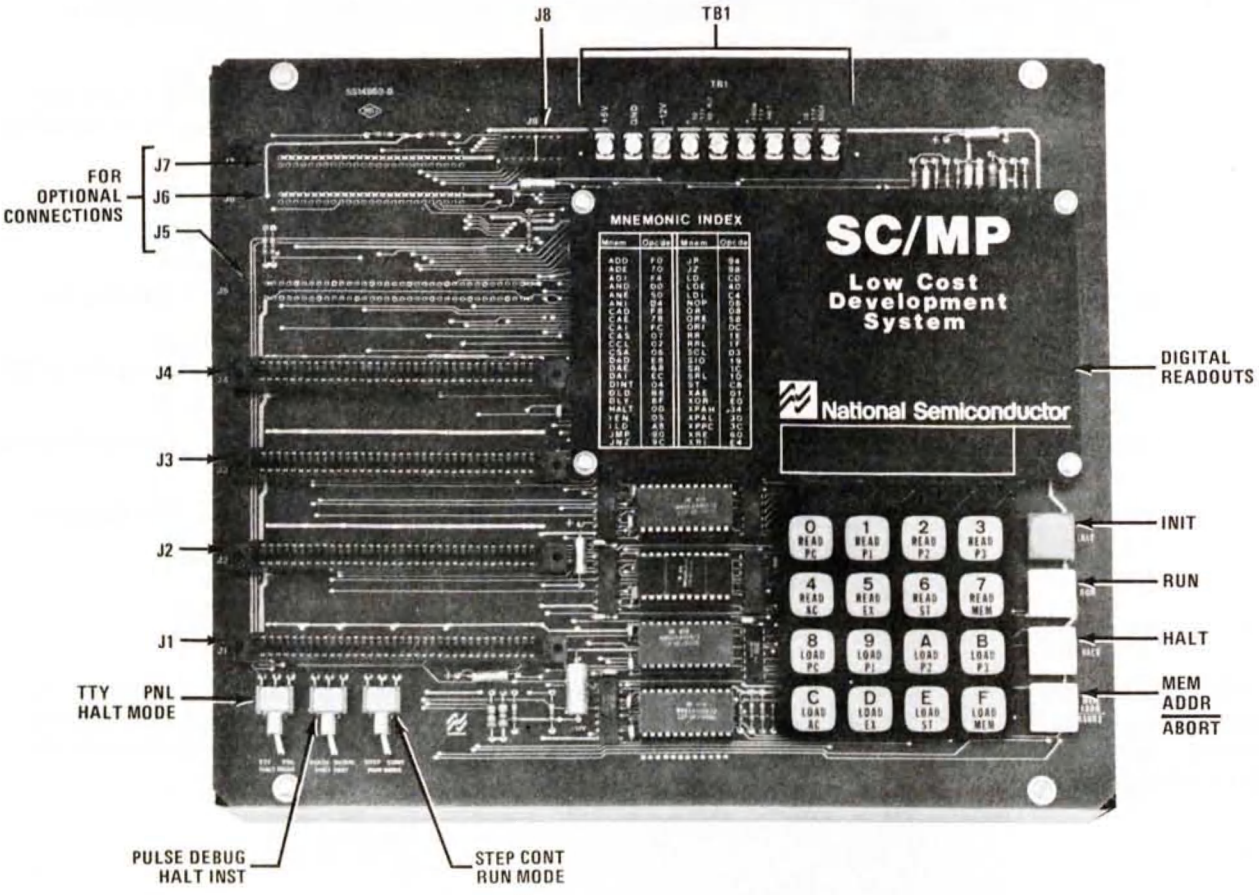
Step	Purpose	Reference
1-4	These are preliminary steps that set up the LCDS for checkout of TTY.	Not Applicable
5	This step verifies that TTY prints out the saved PC address following initialization of SC/MP Microprocessor.	3.2.1
6	This step verifies that Type Address Range Command causes the contents of a selected range of memory addresses to be printed out by TTY.	3.4.5
7	This step verifies that GO command can be used to select RUN Mode operation.	3.4.2
8, 9	These steps verify that GO (Continue) command initiates RUN Mode operation at saved PC address.	3.4.2
10, 11	These steps verify that the Breakpoint Halt command can be used to enter a Halt Instruction at a desired memory address, and that GO (Start Select) command can be used to initiate RUN Mode operation at a specific starting address.	3.4.2
12	This step verifies that Halt Reset command can be used to reset a breakpoint Halt Instruction.	3.4.2
13-16	These steps verify that TTY printout is proper when single-instruction execution of a program is selected.	3.4.2
17-20	These steps verify that contents of a selected range of memory locations can be altered via Alter Address Range command.	3.4.6
21, 22	These steps verify that the Punch command can be used to cause contents of a series of memory locations to be punched into paper tape.	3.4.7
23, 24	These steps verify that the Load command can be used to read a punched tape into memory starting at address specified on tape.	3.4.8
25-28	These steps verify that Alter Memory Address Range command can be used to alter the values saved for SC/MP program counter, registers, and accumulator and that Type Registers command can be used to cause saved values to be printed out on TTY keyboard.	3.4.4
29-31	These steps verify that Load at Specified Address command can be used to read a punched tape into memory starting at memory address specified in command.	3.4.8

7. Visually examine each solder connection for bridges, cold joints, and so forth; if a connection is in doubt, use a continuity checker having an output of 1.5 volts (or less) to test for opens and shorts.

8. Replace motherboard on LCDS chassis and reinstall Phillips-head screws to secure motherboard in place.

9. If applicable, reinstall all tagged external connections to TB1.

10. Replace SC/MP CPU Application Card in motherboard connector.



NS10716

Figure 3-1. LCDS Controls, Connectors, and Indicators

Table 3-1. Description of Controls, Connectors, and Indicators (Continued)

Control, Connector, or Indicator	Description	Reference
READ P2/2 pushbutton	Functional only when LCDS is in DEBUG Mode and Panel Operation is selected, or when addressed by the user's application program while RUN Mode Operation is enabled. When pressed while keyboard is in Command Mode, causes stored P2 value to be displayed on digital readout; when pressed while keyboard is in Numeric Mode, outputs hexadecimal value 2.	3.3.2 and 3.3.3.1
READ P3/3 pushbutton	Functional only when LCDS is in DEBUG Mode and Panel Operation is selected, or when addressed by the user's application program while RUN Mode Operation is enabled. When pressed while keyboard is in Command Mode, causes stored P3 value to be displayed on digital readout; when pressed while keyboard is in Numeric Mode, outputs hexadecimal value 3.	3.3.2 and 3.3.3.1
READ AC/4 pushbutton	Functional only when LCDS is in DEBUG Mode and Panel Operation is selected, or when addressed by the user's application program while RUN Mode Operation is enabled. When pressed while keyboard is in Command Mode, causes stored AC value to be displayed on digital readout; when pressed while keyboard is in Numeric Mode, outputs hexadecimal value 4.	3.3.2 and 3.3.3.1
READ EX/5 pushbutton	Functional only when LCDS is in DEBUG Mode and Panel Operation is selected, or when addressed by the user's application program while RUN Mode Operation is enabled. When pressed while keyboard is in Command Mode, causes stored Extension (E) Register value to be displayed on digital readout; when pressed while keyboard is in Numeric Mode, outputs hexadecimal value 5.	3.3.2 and 3.3.3.1
READ ST/6 pushbutton	Functional only when LCDS is in DEBUG Mode and Panel Operation is selected, or when addressed by the user's application program while RUN Mode Operation is enabled. When pressed while keyboard is in Command Mode, causes stored Status (S) Register value to be displayed on digital readout; when pressed while keyboard is in Numeric Mode, outputs hexadecimal value 6.	3.3.2 and 3.3.3.1
READ MEM/7 pushbutton	Functional only when LCDS is in DEBUG Mode and Panel Operation is selected, or when addressed by the user's application program while RUN Mode Operation is enabled. When pressed while keyboard is in Command Mode, causes address and contents of stored memory reference address to be displayed on digital readout; when pressed while keyboard is in Numeric Mode, outputs hexadecimal value 7.	3.3.2 and 3.3.3.3

Table 3-1. Description of Controls, Connectors, and Indicators (Continued)

Control, Connector, or Indicator	Description	Reference
LOAD MEM/F pushbutton	Functional only when LCDS is in DEBUG Mode and Panel Operation is selected, or when addressed by the user's application program while RUN Mode Operation is enabled. When pressed while keyboard is in Command Mode, enables alteration of stored memory-reference address contents; when pressed while keyboard is in Numeric Mode, outputs hexadecimal value F.	3.3.2 and 3.3.3.4
RUN MODE switch	Associated with RUN Mode operation of LCDS. When set to STEP, enables single-instruction execution of user-entered program; when set to CONTIN enables normal execution of user-entered program.	3.3.1
HALT INST switch	Associated with RUN Mode operation of LCDS. When set to DEBUG, LCDS terminates RUN Mode and enters DEBUG Mode upon execution of Halt Instruction; when set to PULSE, RUN Mode operation is not affected by execution of Halt Instruction.	3.3.1 and 3.4.2
HALT MODE switch	Associated with DEBUG operation of LCDS. When set to PNL, enables DEBUG operational commands to be entered via Panel keyboard; when set to TTY, enables DEBUG operational commands to be entered via TTY keyboard.	3.3.1 and 3.4.2
Connectors J1-J4	Prewired interface connectors that enable plug-in interconnection of SC/MP family Application Cards for hardware- and software-development purposes.	3.5
Connector J5 (optional, user-supplied)	Expands plug-in interface capability for SC/MP family Application Cards.	3.5
Connectors J6 and J7 (optional, user-supplied)	Allows use of flat cable to expand Applications Card Bus for user's system.	3.5
Connector J8	Enables jumper interconnection for various control applications.	3.5.2 and 3.5.5
Terminal Strip TB1	Enables interconnection of primary dc input power and optional TTY.	Chapter 2

Panel Capabilities

- Display contents of SC/MP program counter, registers, and accumulator in hexadecimal format
- Display contents of any memory location in hexadecimal format
- Alter contents of SC/MP program counter, registers, and accumulator
- Alter contents of any memory location
- Initiate execution of user-entered program at any memory address
- Interrupt execution of user-entered program at any point without loss of hardware or software status

Optional TTY

- Print contents of SC/MP program counter, registers, and accumulator
- Print contents of any single memory location, or any selected range of memory locations
- Alter contents of SC/MP program counter, registers, and accumulator
- Alter contents of any single memory location, or any selected range of memory locations
- Set breakpoint halts in Read/Write Memory that enable interruption of user-entered program at any point without loss of hardware or software status
- Initiate execution of user-entered program at any memory address
- Obtain paper tape Load Module by punching selected range of memory locations to paper tape
- Load LCDS generated paper-tape Load Module into memory
- Load IMP-16 or FORTRAN Cross Assembler generated paper-tape Load Module into memory

When the DEBUG Mode is selected, it is entered via a Save Routine that is initiated when the current RUN Mode instruction is completed. The purpose of the Save Routine is to store the contents of the SC/MP program counter, registers, and accumulator in memory locations 77F5 through 77FF (refer to table 3-3) to permit a subsequent return to the RUN Mode without loss of status. During the Save Routine, the contents of the program counter (PC) are incremented by one prior to storage so that they point to the address of the first instruction that will be fetched when the RUN Mode is subsequently selected; the contents of the remaining SC/MP registers and accumulator are not altered by the Save Routine. Upon completion of the Save Routine, the LCDS displays or prints out the stored PC contents to indicate entry into the DEBUG Mode.

Table 3-3. DEBUG Save Addresses for SC/MP Program Counter, Registers, and Accumulator

Memory Address	Stored Value
77F5	Program Counter high-order byte
77F6	Program Counter low-order byte
77F7	Register P1 high-order byte
77F8	Register P1 low-order byte
77F9	Register P2 high-order byte
77FA	Register P2 low-order byte
77FB	Register P3 high-order byte
77FC	Register P3 low-order byte
77FD	Contents of Accumulator
77FE	Contents of Extension Register
77FF	Contents of Status Register

3.3.1 RUN/DEBUG Mode Selection

The RUN Mode is selected by pressing the RUN pushbutton and the DEBUG Mode is selected by default when the RUN Mode is terminated. Panel pushbuttons and switches associated with RUN Mode Operation are RUN, HALT, RUN MODE, and HALT INST. All of the remaining Panel pushbuttons and switches, except the INIT pushbutton, are associated with the DEBUG Mode Operation. Use of the INIT pushbutton is covered in 3.2.1.

To select the RUN Mode of operation, proceed as follows.

NOTE

The following procedure assumes that the user's application program has previously been loaded into memory.

1. Set the RUN MODE switch to CONTIN to enable normal execution of the user's applications program or to STEP to enable single-instruction execution of the user's applications program. When the RUN MODE switch is set to STEP, one instruction will be executed each time that the RUN pushbutton is pressed, and the LCDS will return automatically to the DEBUG Mode. When the RUN MODE switch is set to CONTIN, the user's applications program will be executed continuously while the RUN Mode is enabled.
2. Set the HALT INST switch to DEBUG or PULSE, respectively, to enable or disable DEBUG halting of the user's applications program. When the HALT INST switch is set to DEBUG, execution of an applications program Halt Instruction causes the RUN Mode to be terminated and the DEBUG Mode to be entered. When the HALT INST switch is set to PULSE, RUN Mode Operation is not affected by execution of an applications program Halt Instruction.
3. Press the READ PC pushbutton and observe the PC address displayed on the digital readout. If this is the desired starting address for the user's application program, press the RUN pushbutton to initiate execution of the applications program at that address. If a different starting address is desired, press the LOAD PC pushbutton and key in the desired address. Then, press the RUN pushbutton to initiate program execution.

After the RUN pushbutton is pressed, the LCDS digital readout will remain blanked until the RUN Mode is terminated and the DEBUG Mode is entered. (Upon entry to the DEBUG Mode, the LCDS will display the saved PC address on the digital readout.) Termination of the RUN Mode and entry into the DEBUG Mode will occur under any of the following conditions.

1. The INIT pushbutton is pressed (refer to 3.2.1).
2. The HALT pushbutton is pressed. (When the HALT pushbutton is pressed, termination of the RUN Mode will occur when the instruction in progress is completed.)
3. The RUN/STEP switch is set to STEP, and one instruction is executed.
4. The HALT Mode switch is set to DEBUG, and a Halt Instruction is executed.
5. The DEBUG input to the LCDS is externally driven low (refer to 3.5.4).

3.3.2 DEBUG Commands

When the LCDS is in the DEBUG Mode, the keyboard pushbuttons on the Panel function under control of the DEBUG firmware to provide either command or numeric outputs. The command outputs of the pushbuttons are indicated by the screened legends (READ PC, READ P1, etc.) and the numeric outputs are indicated by the screened hexadecimal values (0, 1, 2, and so forth). Operation of the DEBUG firmware is such that the command outputs of the pushbuttons are always enabled except when a numeric value must be entered to complete a command (LOAD PC, LOAD P1, and so forth). When this occurs, the DEBUG firmware enables the numeric outputs of the pushbuttons, and the LCDS prompts for input by displaying hyphens (---- or --) in the appropriate digits of the digital readout. Pressing the keyboard switches then causes hexadecimal values to be entered (left to right) and displayed on the appropriate digits of the digital readout. When the last value is entered, the DEBUG firmware automatically executes the command and reenables the command outputs

Pushbutton Pressed	Prompt Indication
LOAD PC	---- PC
LOAD P1	---- P1
LOAD P2	---- P2
LOAD P3	---- P3
LOAD AC	-- A
LOAD EX	-- E
LOAD ST	-- S

3.3.3.3 Examining Contents of a Memory Location

The contents of a memory location may be examined either via the MEM ADDR/ABORT pushbutton or via the READ MEM pushbutton. Operation of the pushbuttons is as follows.

1. When the MEM ADDR/ABORT pushbutton is pressed while no command is in progress, the LCDS prompts for a 4-digit address input by displaying hyphens (----) on the digital readout. After the desired 4-digit address is keyed in, the contents of that memory location are displayed automatically on the last two digits of the digital readout.

2. When the READ MEM switch is pressed, the DEBUG firmware refers to a stored memory-reference address to determine which memory location was selected for display. The address of the selected memory location then is displayed on the first four digits of the digital readout, and the contents of the memory location are displayed on the last two digits. During DEBUG operation, the DEBUG firmware automatically sets the stored memory-reference address to the last 4-digit hexadecimal value displayed on the digital readout. In addition, each time that the contents of a memory location are examined via the MEM ADDR/ABORT or READ MEM pushbuttons, the DEBUG firmware also enables a READ MEM auto-increment loop. The auto-increment loop then remains enabled until some other command is entered (READ PC, READ AC, LOAD MEM, or another). Thus, if the last command processed was MEM ADDR or READ MEM, pressing the READ MEM pushbutton will cause the stored memory-reference address to be incremented by one before the current READ MEM command is executed; if the READ MEM pushbutton is pressed following any other command, the memory location accessed will correspond to the last 4-digit hexadecimal value previously displayed on the digital readout. Examples of using the READ MEM pushbutton are provided below.

Command Sequence	Digital Readout Indication
INIT	0001 PC
READ P1	0000 P1
READ MEM	0000 "XX"
READ MEM	0001 "XX"
READ MEM	0002 "XX"
READ AC	00 A
READ MEM	0002 "XX"
READ P1	0000 P1
MEM ADDR (enter 7800)	---- / 7800 C8
READ MEM	7801 C4
READ MEM	7802 08
READ P2	0000 P2
READ MEM	0000 "XX"
LOAD P2 (enter 7810)	---- P2 / 7810 P2
READ MEM	7810 31

Table 3-4. TTY DEBUG Command Notations and Symbols

Notation/Symbol	Meaning
Underscored characters, numbers, and symbols (XXXX)	Underscored characters, numbers, and symbols indicate inputs that are entered by the user at the TTY keyboard. For example: - T 100 (DEBUG firmware prompt/user input) - T 200:204 (DEBUG firmware prompt/user input)
Non-underscored characters, numbers, and symbols (XXXX)	Non-underscored characters, numbers, and symbols indicate outputs from the DEBUG firmware that are printed by the TTY. For example: - T 100 CR (DEBUG firmware prompt/user input) 0100 XX (DEBUG firmware output, output undefined) - T 200:203 (DEBUG firmware prompt/user input) 0200 hv hv hv hv (DEBUG firmware output and prompt for next command)
hv	Denotes a 2-digit hexadecimal value that is typed in or printed out via the TTY keyboard. If more than the desired number of digits are typed in, the DEBUG firmware only accepts the last two digits. Leading zeroes may be omitted or included, as desired.
ma	Denotes a 4-digit hexadecimal value that specifies a memory address. If more than four digits are typed in, the DEBUG firmware only accepts the last four digits. Leading zeroes may be omitted or included, as desired.
hvr	Denotes a value list comprised of 2-digit hexadecimal values separated by commas. Any digits omitted between commas are interpreted as zeroes. Thus, A would be interpreted as 0A, 00. NOTE A memory-address range must be located on a single "page" or "wraparound" will occur; refer to 3.4.1, TTY DEBUG Commands.
mar	Denotes a memory-address range consisting of a memory address (ma), a colon (:), and a second memory address (ma); that is, mar = ma:ma. All locations from the first entry through the last are included in the range. For example, 001A:02FF signifies all memory locations from 001A through 02FF including 001A and 002F. The memory address to the left of the colon represents the low limit of the range, and the address to the right represents the high limit. If the upper limit of the range is smaller than the lower limit, the DEBUG firmware accepts only the left number and executes the specified command at that address.
comm	Denotes comment consisting of English-language text, including letters and numbers. Associated with TTY utilities covered under 3.6, Program Development on LCDS.
, (comma)	Used during entry of commands to separate hexadecimal values (hv) from one another, or from another element of the command statement. When commas are used to separate hexadecimal values, any digits omitted are treated as zeroes; thus, AA,,BB would be treated as AA, 00, BB.

NOTE

When an SC/MP Pointer Register (PC, P1, P2, or P3) is incremented, only the 12 low-order bits are affected; the 4 high-order bits remain at the value to which they were previously set via initialization of the SC/MP microprocessor, or via an XPPC, XPAH, or Transfer (JMP, JP, JZ, and JNZ) Instruction. The term 'page', therefore, is used in the following paragraphs to indicate the group of 4,096 memory locations specified by the value of the Pointer Register four high-order bits (that is, X000 through XFFF). For reference purpose, SC/MP page designations are listed in table 3-6.

The TTY Type Register Range and Punch DEBUG commands permit a range of memory addresses to be specified in the command. The only restriction on specifying a range is that the entire range must be contained within a single page to prevent "wraparound" from occurring. During execution of commands that specify a range, the DEBUG firmware employs an SC/MP Pointer Register as a reference counter (1) by setting the Pointer Register to the appropriate starting address and, then, (2) by automatically incrementing the Pointer Register to permit printout of the specified memory data. Wraparound will occur, therefore, if the Pointer Register is incremented through a page boundary; that is, if the Pointer Register reaches the end of the page to which it has been previously set, it will automatically loop back to the start of the same page and thereby cause an erroneous printout. For example, if memory addresses, 3FFE through 4001 were specified, the printout would actually indicate the contents of memory locations 3FFE, 3FFF, 3000, and 3001.

The Alter Address Range Command permits modification of data in a range of memory locations, so, again, the entire range must be on a single page to prevent wraparound from occurring during automatic incrementing of the SC/MP Pointer Register used as a reference counter. In addition, since new instructions or data values are entered via this command, the user must ensure that the existing Pointer Register page boundaries are observed within the applications program to prevent wraparound from occurring during PC-Relative, Indexed, or Auto-Indexed addressing. Detailed information on PC-Relative, Indexed, and Auto-Indexed addressing is provided in the SC/MP Programming and Assembler Manual.

The Load at Specified Address command causes a paper-tape Load Module to be read into contiguous memory locations starting at the address specified in the command instead of the address specified on the tape. The user must take care, therefore, to ensure that the starting address does not cause wraparound to occur during loading of the tape or during future execution of the applications program.

Table 3-5. TTY DEBUG Commands

Command	Format	Description
		TYPE COMMANDS
Type Address	T ma	This command causes the address of the specified memory location to be printed out in 4-digit hexadecimal notation, and the contents of the memory location to be printed out in a 2-digit hexadecimal notation. For example: - T 200 CR (DEBUG firmware prompt/user input) 0200 hv (DEBUG firmware output and prompt for - / next command) (contents of 0200)

Table 3-5. TTY DEBUG Commands (Continued)

Command	Format	Description
BREAKPOINT COMMANDS		
Breakpoint Halt	H ma	This command enters a Halt Instruction into the specified memory location after saving the original contents of the memory location. For example typing - H 100 CR would cause a Halt Instruction to be stored at memory location 0100. Only one breakpoint halt may be in effect at one time; if a second breakpoint halt command is entered, the DEBUG firmware resets the first breakpoint halt by restoring the original contents to the memory location used for the first breakpoint halt. The Breakpoint Halt command will work properly only when the HALT INST switch is set to DEBUG.
Breakpoint Halt Reset	H	This command cancels a previous breakpoint halt command without setting a new breakpoint halt. Cancellation of the previous breakpoint halt command is effected by restoring the original contents to the memory location used for the previous breakpoint halt.
GO COMMANDS		
GO (Continue)	G	This command causes RUN Mode operation to be initiated at the PC address stored in memory locations 77F5 and 77F6.
GO (Start Select)	G ma	This command specifies the address at which RUN Mode operation will be initiated. For example, typing - G 100 will cause RUN Mode operation to be initiated at address 0100.
PUNCH COMMAND		
		NOTE The memory-address range specified by the Punch command must be located on a single memory "page" or "wraparound" will occur; refer to 3.4.1, TTY DEBUG Commands.
Punch	P mar	This command causes the contents of the specified memory locations to be punched into paper tape in SC/MP Load Module Format (refer to SC/MP Programming and Assembler Manual). The contents of the specified memory locations are not altered.
LOAD COMMANDS		
Load at Tape Address	L	This command enables an LM tape to be loaded into memory at the addresses as specified on the tape. NOTE The starting address selected via the Load at Specified Address command must permit the entire tape to be loaded on a single "page" or "wraparound" will occur; refer to 3.4.1, TTY DEBUG Commands.
Load at Specified Address	L ma	This command causes the LCDS to ignore the starting address specified on the LM tape and, instead, loads the LM tape into memory in contiguous locations starting at the address specified in the command.

3.4.2 RUN/DEBUG Mode Selection

The RUN Mode is selected by typing - G (CR) or G ma (CR) on the TTY, and the DEBUG Mode is selected by default when the RUN Mode is terminated. LCDS switches and TTY DEBUG commands associated with RUN Mode operation are the INIT pushbutton, the HALT switch, the HALT INST switch, the RUN Continue Command, the RUN Start Select Command, the Breakpoint Halt Command, and the Breakpoint Halt Reset Command. Use of the INIT pushbutton is covered under 3.2.1, Power-Up and Initialization. Use of the remaining switches and TTY commands are covered in the instructions that follow.

To select the RUN Mode of operation, proceed as follows.

1. Set RUN Mode switch to CONTIN to enable normal execution of applications program or to STEP to enable single-instruction execution of the applications program. When RUN MODE switch is set to CONTIN, applications program will be executed continuously until RUN Mode is terminated. When RUN Mode switch is set to STEP, one instruction will be executed each time that RUN Mode is selected; then, LCDS will return automatically to DEBUG Mode.

2. Set HALT INST switch to DEBUG or PULSE, respectively, to enable or disable DEBUG halting of applications program. When HALT INST switch is set to DEBUG, RUN Mode will be terminated and DEBUG Mode will be entered whenever a Halt Instruction is executed by applications program. When HALT INST switch is set to PULSE, RUN Mode Operation is not affected by execution of an applications program Halt Instruction. A Halt Instruction may be temporarily entered at desired applications program address via TTY Breakpoint Halt Command. The Halt Instruction then will remain in effect until either a second Breakpoint Halt Command or a Breakpoint Halt Reset Command is executed. Since only one Breakpoint Halt Command may be in effect at one time, each succeeding Breakpoint Halt Command resets previous Breakpoint Halt Command by restoring original contents of memory location at which Halt Instruction was stored temporarily. When a Breakpoint Halt Reset Command is executed, last Breakpoint Halt Command is reset by restoring original contents to appropriate memory location without loading a Halt Instruction into a new memory location. To enter a Breakpoint Halt Command, type

   - H ma (CR) (for example, typing - H 100 (CR) would cause a Halt Instruction to be stored at memory location 0100)

   To enter a Breakpoint Halt Reset Command, type

   - H (CR)

3. Type - TR (CR) and observe resultant printout on the TTY. If PC value indicated is desired starting address for application program, type - G (CR) to initiate RUN Mode Operation at that address. If a different starting address is desired, type - G ma (CR) to initiate RUN Mode Operation at address specified by ma value.

After the RUN Command is entered, the TTY keyboard will be disabled until the DEBUG Mode is subsequently entered. Upon entry into the DEBUG Mode, the TTY will print out: - CL hv to indicate the saved PC value. Termination of the RUN Mode and entry into the DEBUG Mode will occur under any of the following conditions.

1. The INIT switch is pressed (refer to 3.2.1, Power-up and Initialization).
2. The HALT switch is pressed (RUN Mode termination occurs after the instruction in progress is completed).
3. The RUN/STEP switch is set to STEP. (RUN Mode termination occurs after one instruction is executed.)
4. The HALT INST switch is set to DEBUG and a Halt Instruction is executed.
5. The DEBUG input to the LCDS is externally driven low (refer to 3.5.4, DEBUG* Signal Implementation).

3.4.8 Loading a Paper Tape

To read a paper tape into memory, proceed as follows:

1. Raise cover on Tape Reader, install tape on sprocket, and then close cover to hold tape in place.
2. If it is desired to load the tape into memory using addresses specified on the tape, type - L (CR). If it is desired to load into memory starting at some other address, type - L ma (CR). (For example, typing - L 100 (CR) would cause the tape to be loaded into memory starting at address 0100.)

NOTE

No adjustment of the object code is attempted. The data on the tape are merely read into contiguous memory locations starting at the specified address.

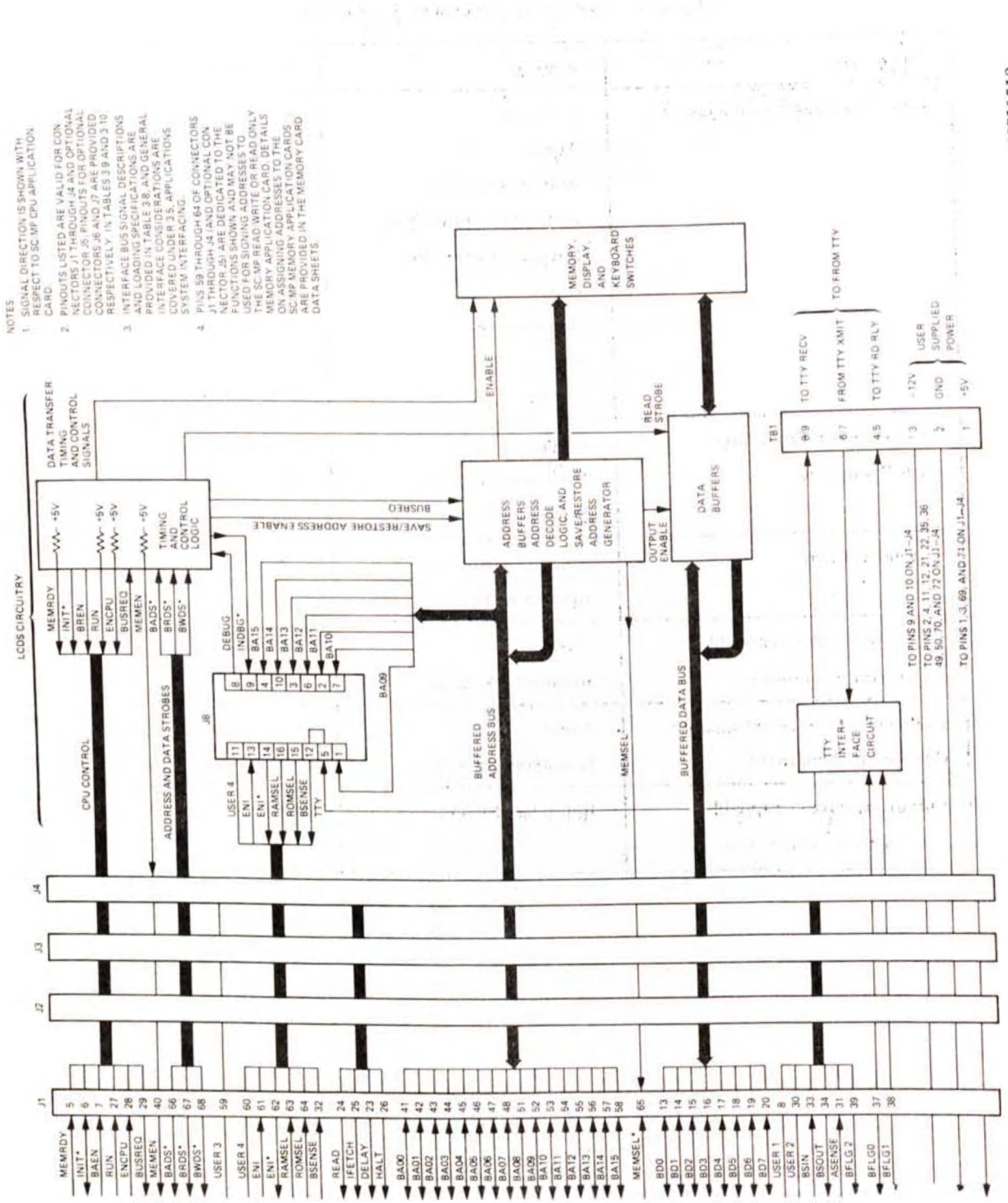
3. Set Tape Reader to START and observe that tape feeds through. When tape stops, remove tape and set Tape Reader to STOP.

3.5 APPLICATION SYSTEM INTERFACING

The LCDS interface bus (figure 3-2) is prewired to permit an SC/MP CPU Application Card to function as part of the user's application system when the LCDS is in the RUN Mode, and as part of the LCDS when the LCDS is in the DEBUG Mode. The way that the LCDS accomplishes this is by reserving addresses 7000 through 7FFF for DEBUG operations and by permitting the remainder of the 64K addresses to be assigned as desired by the user. From a software standpoint, therefore, the RUN Mode covers the address range: 0000-6FFF and 8000-FFFF.

For definition purposes, the term "user's application system" is considered to include all circuits that provide an operational interface with the SC/MP CPU Applications Card when the LCDS is in the RUN Mode. Thus, for small-scale applications, the user's application system may be nothing more than the 256-by-8-bit Read/Write Memory on the SC/MP CPU Application Card. For large-scale applications, the user's application system may consist of a variety of SC/MP family application cards, peripherals, and/or special-purpose custom memory and control circuits.

Connectors J1 through J4 provide a 72-pin plug-in interface for SC/MP family application cards (all cards must be installed with component side facing front of LCDS). If desired, one more 72-pin connector may be installed at socket location J5 to accommodate an additional SC/MP family application card, and ribbon connectors may be installed at motherboard locations J6 and J7 to accommodate custom applications system designs (refer to chapter 2, Installation). Sources of accessory equipment useful for fabricating and interconnecting custom application cards are listed in table 3-7.



NS10512

Figure 3-2. Application System Interface Bus

Table 3-8. SC/MP Application Card Interface Bus Description (Continued)

Pin	Signal	CPU Output at BADS* Time		CPU Input at BRDS* Time	CPU Output at BWDS* Time
		Functional Name	Description		
13	BD0	BA12	Fourth most significant bit of 16-bit address.	↑	↑
14	BD1	BA13	Third most significant bit of 16-bit address.		
15	BD2	BA14	Second most significant bit of 16-bit address.		
16	BD3	BA15	Most significant bit of 16-bit address		
17	BD4	READ	When high, data input cycle is starting; when low data output cycle is starting	Input data are expected on the eight (BD0-BD7) lines.	Output data are valid on the eight (BD0-BD7) lines.
18	BD5	IFETCH	When high, first byte of instruction is being fetched; processed by LCDS to control entry into DEBUG Mode.		
19	BD6	DELAY	When high indicates that delay cycle is starting; that is, second byte of Delay Instruction is being fetched.		
20	BD7	HALT	When high, indicates that Halt Instruction has been executed; causes LCDS to terminate RUN Mode and enter DEBUG Mode when HALT INST switch is set to DEBUG.	↓	↓

Pin	Signal	Description
21, 22	GND	Power and signal ground for SC/MP Application Cards.
23	DELAY	Latched Delay Status Flag output of SC/MP CPU Application Card. Latched high on leading edge of BADS* strobe to indicate that delay cycle is starting (that is, second byte of Delay Instruction is to be fetched during data input/output cycle in progress); otherwise, latched low on leading edge of BADS* strobe. Not used by LCDS or SC/MP Memory Application Cards.
24	READ	Latched Read Status Flag output of SC/MP CPU Application Card. Latched high or low, respectively, on leading edge of BADS* strobe to indicate whether read or write data input/output cycle is in progress. Not used by LCDS or SC/MP Memory Application Cards.

Table 3-8. SC/MP Application Card Interface Bus Description (Continued)

Pin	Signal	Description
38	BFLG1	Buffered General-Purpose Flag Output of SC/MP CPU Application Card. During DEBUG Mode, used by LCDS to control TTY tape-reader relay. Not used by SC/MP Memory Application Cards.
39	BFLG2	Buffered General-Purpose Flag Output of SC/MP CPU Application Card. Not used by LCDS or SC/MP Memory Application Cards.
40	MEMEN	Address Compare Enable input to SC/MP Memory Application Cards. When high, enables addressing of SC/MP Memory Application Cards; when low, inhibits addressing of SC/MP Memory Application Cards. Pulled up to +5 volts via resistor on LCDS. Not used by SC/MP CPU Application Card.
41	BA00	Buffered 16-bit Address Bus, which is set to high-impedance state except when data input/output cycle is in progress. Used by SC/MP CPU Application Card to communicate with application system or LCDS memory except during DEBUG Save and Restore Routines. During DEBUG Save and Restore Routines, LCDS sets BAEN Signal low to disable address output of SC/MP CPU Application Card, and forces addresses onto bus to control DEBUG Mode entry and exit. SC/MP Memory Application Cards have internal comparator circuit that continually compares address on bus with address assigned to card to enable or disable card, as appropriate.
42	BA01	
43	BA02	
44	BA03	
45	BA04	
46	BA05	
47	BA06	
48	BA07	
49	See below	
50	See below	
51	BA08	
52	BA09	
53	BA10	
54	BA11	
55	BA12	
56	BA13	
57	BA14	
58	BA15	
49, 50	GND	Power and signal ground for SC/MP application cards. NOTE LCDS interface connectors J1 through J4 are prewired to enable installation of SC/MP CPU Application Card at any location. Thus, pins 61 through 64 are dedicated to the functions listed below and may not be used for externally assigning addresses to SC/MP Memory Application Cards. Details on assigning addresses to SC/MP Memory Application Cards via on-card jumper options are provided in memory card data sheets.
59	USER 3	Provided for user implementation in special-purpose applications; not used by LCDS or SC/MP CPU Application Cards.
60	USER 4	Provided for user implementation in special-purpose applications; not used by LCDS or SC/MP Memory Application Cards. Also available at LCDS connector J8 (refer to table 3-2).
61	EN1	SC/MP CPU Application Card input/output signals typically used for on-card memory control. (EN1 input is inverted to provide EN1* output.) Both signals are also accessible at LCDS connector J8 (refer to table 3-2).
62	EN1*	

Table 3-9. Ribbon Connector J6 (Optional) Pin Assignments

Pin	Signal	Refer to Table 3-8, Pin:
1	GND	2
2	BRDS*	67
3	GND	2
4	MEMSEL*	65
5	GND	2
6	USER 3	59
7	GND	2
8	BA14	57
9	GND	2
10	BA12	55
11	GND	2
12	BA10	53
13	GND	2
14	BA08	51
15	GND	2
16	BA06	47
17	GND	2
18	BA04	45
19	GND	2
20	BA02	43
21	GND	2
22	BA00	41
23	GND	2
24	BFLG2	39
25	GND	2

Pin	Signal	Refer to Table 3-8, Pin:
26	BFLG0	37
27	GND	2
28	BSOUT	34
29	GND	2
30	BSENSE	32
31	GND	2
32	USER 2	30
33	GND	2
34	ENCPU	28
35	GND	2
36	HALT	26
37	GND	2
38	READ	24
39	GND	2
40	BD7	20
41	GND	2
42	BD5	18
43	GND	2
44	BD3	16
45	GND	2
46	BD1	14
47	GND	2
48	USER 1	8
49	GND	2
50	INIT*	6

3.5.2 SC/MP CPU Application Card Memory Control

The SC/MP CPU Application Card contains a 256-by-8-bit read/write memory and has provision for installation of an optional 512-by-8-bit read-only memory. Selection of the read/write or the read-only memory is effected, respectively, by the RAMSEL and ROMSEL inputs to the card as shown in figure 3-3. Thus, when the read/write memory or the read-only memory is to serve as part of the user's applications system during RUN Mode operation, address decoding may be necessary for proper control of the RAMSEL and ROMSEL signals. Address decoding options for RAMSEL and ROMSEL control are listed in table 3-11.

3.5.3 Application System Address Assignment

The address ranges available for assignment to the user's application system are 0000-6FFF and 8000-FFFF. If the user's application system includes SC/MP memory application cards that are installed in connector(s) J1 through J4, the internal jumper options on the memory cards must be used for address assignment and pins 59 through 64 of the memory cards must be left unterminated to permit normal operation of the SC/MP CPU Application Card. Instructions for assigning addresses to the memory cards via the internal jumper options are provided in the data sheets for the SC/MP Read/Write Memory and the SC/MP ROM/PROM Memory Application Cards.

For SC/MP application systems based on discrete components or custom interconnection of the SC/MP memory applications cards, the method employed for address assignment is left to the discretion of the user since a variety of schemes are possible (refer to SC/MP Technical Description, publication number 4200079).

NOTE

As shown in figure 3-3, address bits BA09 through BA15 are brought out to LCDS connector J8 to provide an easy access for the user.

3.5.4 DEBUG* Signal Implementation

For various applications system requirements it may be desirable to detect the occurrence of an asynchronous event and to monitor the operating conditions that were in effect at the time of the event. This can be readily accomplished by using the DEBUG* input to the LCDS (available at pin 8 of connector J8). When the DEBUG* signal goes low while the LCDS is in the RUN Mode, the RUN Mode is terminated and the DEBUG Mode is entered after the instruction in progress is completed. Thus, external circuits can be employed to drive the DEBUG* signal low in response to an external event; then, the information desired can be obtained by examining the contents of memory locations and/or the values stored for the SC/MP program counter, registers, and accumulator. While the DEBUG* signal remains low, the RUN Mode Operation is inhibited; the DEBUG* signal must be reset high, therefore, before RUN Mode operation can be reenabled via the RUN switch or the TTY RUN Commands.

If the DEBUG* signal goes low while the LCDS is in the DEBUG Mode, it will not affect LCDS operation except for inhibiting selection of the RUN Mode while it remains low.

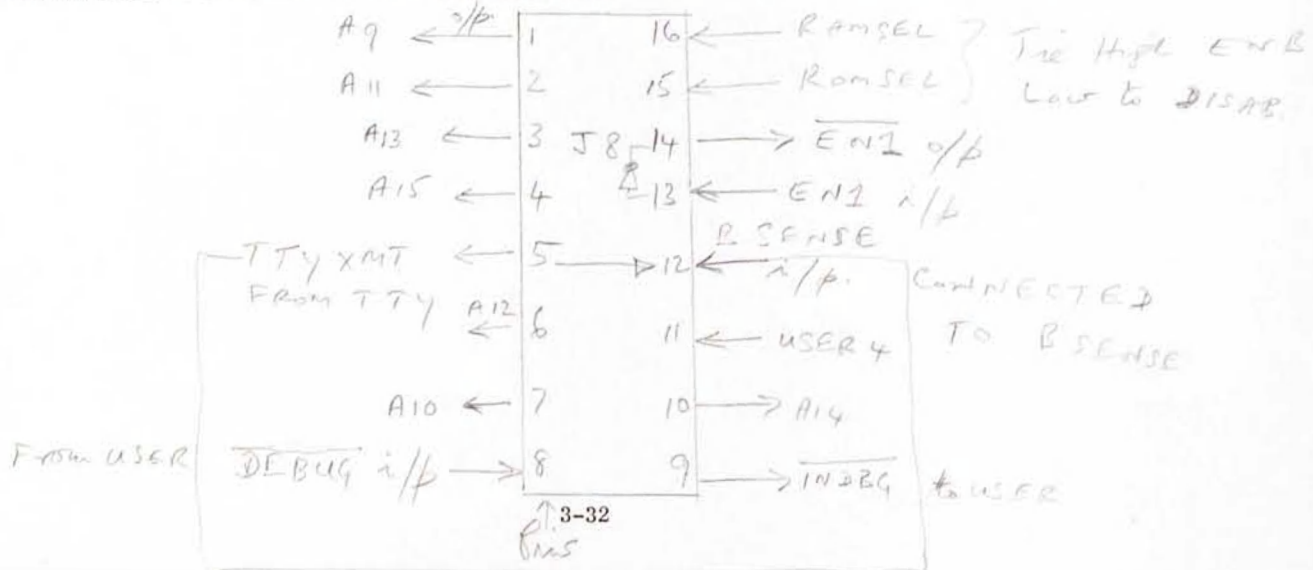


Table 3-11. SC/MP CPU Application Card Memory Control Options

Conditions	Decoding Required for RAMSEL and ROMSEL Signals
1a. Read-only memory is not installed on SC/MP CPU Application Card. 1b. Read/write memory on SC/MP CPU Application Card is used as stand-alone applications system or as part of applications system that consists exclusively of SC/MP application cards.	No special decoding is required for the RAMSEL and ROMSEL signals, but the RAMSEL signal should be tied to +5V via an 1-kilohm resistor to ensure a logic '1' input to the SC/MP CPU Application Card. For the conditions specified, the Read/Write Memory on the SC/MP CPU Application Card is controlled by the MEMSEL* input. When either the LCDS or the memory application cards are addressed, the MEMSEL* signal will be low and the read/write memory on the SC/MP CPU Application Card will be disabled; when neither the LCDS nor the memory application cards are addressed, the MEMSEL* signal will be high and the Read/Write Memory on the SC/MP CPU Application Card will be enabled.
2a. Read-only memory is not installed on SC/MP CPU Applications Card. 2b. Read/write memory on SC/MP CPU Applications Card is used as part of applications system that contains custom memory and/or peripheral circuits in addition to or in lieu of SC/MP memory application card(s).	If the custom application system circuit is configured to provide a low-going MEMSEL* output to the SC/MP CPU Applications Card while addressed, no special decoding of the RAMSEL and ROMSEL signals is necessary, but the RAMSEL signal should be pulled up to +5V via a 1-kilohm resistor to ensure a logic '1' input to the SC/MP CPU Application Card. If the custom applications system circuit is not configured to provide a low MEMSEL* output while addressed, Address Bits BA09-BA15 can be decoded such that RAMSEL is driven high for a unique 256-address range and low at all other times, or RAMSEL can be tied to an address bit that is out of the address range of the applications system when RAMSEL goes high. For example, if the highest address assigned to the applications system were 4FFF, RAMSEL could be jumpered to address bit BA15. The low-order address of the SC/MP CPU Application Card Read/Write Memory then would be 8000.
3a. Read-only memory is installed on SC/MP CPU Applications Card. 3b. Read-only memory on SC/MP CPU Application Card is used as stand-alone applications system or as part of applications system that consists exclusively of SC/MP application cards. 3c. Read/write memory on SC/MP CPU Application Card is not used as part of applications system.	RAMSEL should be tied to ground and ROMSEL should be pulled up to +5V via a 1-kilohm resistor. The Read/Write Memory on the SC/MP CPU Application Card then will be held disabled, and the read-only memory will be controlled by the MEMSEL* input as described for condition 1.

3.5.5 RUN Mode Considerations

Since the LCDS is primarily designed to permit the design of a user's application system to be tested in an operational environment, it enables all of the operating capabilities of the SC/MP CPU Application Card to be utilized during the RUN Mode Operation. For certain types of applications, however, special processing of the signals listed below may be necessary to ensure that the applications system does not interfere with LCDS operation during the DEBUG Mode, and the LCDS does not interfere with applications system operation during the RUN Mode.

1. BFLG0 and BFLG1 Signals. During DEBUG operation, the BFLG0 and BFLG1 outputs of the SC/MP CPU Application Card are used, respectively, to transmit data to the optional TTY and to control the TTY Reader Relay. Thus, if these signals are to be used also by the applications system during RUN Mode Operation, they should be applied to the applications system via a gate that is controlled by the INDBG* status output of the LCDS (see figure 3-4).

   NOTE

   When the DEBUG Mode is entered, the contents of the SC/MP Status Register are saved in memory location 77FF. Thus, the logical states of BFLG0 and BFLG1 will be saved when the RUN Mode is terminated and reinstated when the RUN Mode subsequently is selected.

2. BSENSE Signal. During DEBUG Mode Operation, the BSENSE input to the SC/MP CPU Application Card is used for reception of data from the optional TTY. Thus, if the BSENSE signal is also to be used by the applications system during RUN Mode Operation, the jumper installed at the factory between pins 5 and 12 of connector J8 (refer to table 3-2) should be disconnected, and a single-throw switch should be installed instead. The switch then can be set to OFF to disconnect the BSENSE and TTY signals during RUN Mode Operation, and to ON to reconnect the signals during DEBUG Mode Operation.

3. Interrupts. If the SC/MP Internal IE Flag is set during RUN Mode Operation to enable interrupts, the ASENSE/Interrupt signal should be applied to the SC/MP CPU Applications Card via a gate that is controlled by the INDBG* status output of the LCDS (see figure 3-4).

3.6 PROGRAM DEVELOPMENT ON LCDS

For detailed information on SC/MP program development and machine-language coding of SC/MP instructions, refer to the SC/MP Programming and Assembler Manual (publication number 4200094). The paragraphs that follow describe the use of special features designed into the LCDS to simplify applications program development and debug operations.

3.6.1 TTY Utilities

The LCDS DEBUG firmware contains a number of TTY utility subroutines that can be called by the user's application program to send and receive data to and from the TTY keyboard and the TTY tape reader. A brief description of each of the subroutines available to the user is provided in table 3-12 along with the subroutine calling address.

Detailed instructions for using the subroutines in an applications program are provided in the paragraphs that follow and functional operation of the subroutines is shown in figures 3-5 through 3-9. The following general conventions are associated with all of the subroutines.

1. Input Data. All of the subroutines associated with receiving data from the TTY keyboard and/or the TTY tape reader employ the BSENSE input to the SC/MP Microprocessor as the input data line. When data are to be received from the TTY keyboard only, the Flag 1 output of the SC/MP Microprocessor is reset low to deenergize the TTY Reader Relay; when data are to be received from the TTY tape reader, the Flag 1 output of the SC/MP Microprocessor is set high to enable the TTY Reader Relay.

2. Output Data. All of the subroutines associated with transmitting output data to the TTY employ the Flag 0 output of the SC/MP Microprocessor as the output data line.

3. Software Stack. All of the subroutines reference a SC/MP Pointer Register 2 (P2) as a stack pointer for development of a software stack that grows from high to low memory. Thus, before a subroutine is called P2 must be set to the high-order address of the stack. During execution of the subroutine, the stack then is used for saving and restoring SC/MP registers and for temporary storage of working and input/output data. The maximum stack length required is seven memory locations. The following instruction sequence may be used to load a 4-digit memory address into P2:

```
LDI     hv      Load AC with low-order P2 address
XPAL    2       Exchange P2-low with AC
LDI     hv      Load AC with high-order P2 address
XPAH    2       Exchange P2-high with AC
```

4. Subroutine Calls and Returns. All of the subroutines save the contents of the SC/MP Pointer Register 3 (P3) upon entry; subsequently, the subroutines effect a return to the calling program, first, by restoring the saved contents to P3 and, then, by exchanging the contents of P3 with the contents of the Program Counter. Thus, to ensure a proper return, a subroutine should be called, first, by loading the appropriate address into P3 and, second, by exchanging the contents of P3 with the contents of the Program Counter via an XPPC 3 instruction. The following instruction sequence may be used to load a 4-digit memory address into P3:

```
LDI     hv      Load AC with low-order P3 address - 1
XPAL    3       Exchange P3-low with AC
LDI     hv      Load AC with high-order P3 address
XPAH    3       Exchange P3-high with AC
```

NOTE

To facilitate use, all of the subroutine addresses in this chapter specify the low-order P3 address - 1 value; for all subroutine calls, therefore, load the appropriate address into the P3 register exactly as shown.

5. Interrupts. In order for the SC/MP Microprocessor to process interrupts, it is necessary that P3 be used as an interrupt-service-routine pointer. Thus, while P3 is used as a subroutine pointer, interrupts must be disabled by programming the SC/MP Interrupt Enable (IE) Flag to the reset state.

Since an XPPC 3 Instruction is employed by each of the subroutines to reinstate the PC return address, P3 always points to the exit address of the subroutine upon return to the applications program. Subroutines that can be recalled via the exit address are considered repeatable; subroutines that cannot be recalled via the exit address are considered nonrepeatable. The repeatable subroutines are GECHO, PUTC, MESG, GHEX, and PHSEB. The remaining subroutines (GETP, GETC, PECHO, GHEXE, and PHEX) are not repeatable because their exit addresses are common to one of the repeatable subroutines (see figures 3-5 through 3-9). In the subroutine descriptions that follow, both initial and repeat calling sequences are listed for each subroutine.

Table 3-13. USA Standard Code for Information Interchange (ASCII)

b_6 b_5 b_4 → / BITS ↓					0_0	0_1	0_1	0_1	1_0	1_0	1_1	1_1
b_3 ↓	b_2 ↓	b_1 ↓	b_0 ↓	Column→ / Row ↓	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	|
1	1	0	1	13	CR	GS	-	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Note: The hex value = Column, Row; for example, A = 41, B = 42, Q = 51, and so forth.

3.6.1.1 GETC Subroutine

The GETC Subroutine (figure 3-5), first, sets the SC/MP Status Register to 00 to disable the TTY tape reader, and, then, reads one character into the SC/MP Accumulator from the TTY keyboard without echo. Execution of GETC, therefore, causes the original contents of the SC/MP Accumulator and the Status Register to be lost. The contents of the remaining SC/MP registers, except for P3, are not affected.

Before GETC may be called initially, P2 must be set to the high-order address of a software stack that allows at least five memory locations for use by GETC, and P3 must be set to the GETC call address 7A88. After P2 and P3 are set to the appropriate addresses, GETC may be called via an XPPC 3 Instruction. Upon completion, GETC will return to the next sequential instruction of the applications program with the received character stored in the SC/MP Accumulator, and the SC/MP Pointer Register 3 (P3) set to the exit address GECHO. Thus, if GETC is to be recalled to permit reception of another character from the TTY keyboard, the contents of the Accumulator, first, must be stored in a memory location to permit future processing, and, then, P3 must be reloaded with the GETC call address 7A88. Following these two actions, GETC may be recalled via an XPPC 3 instruction.

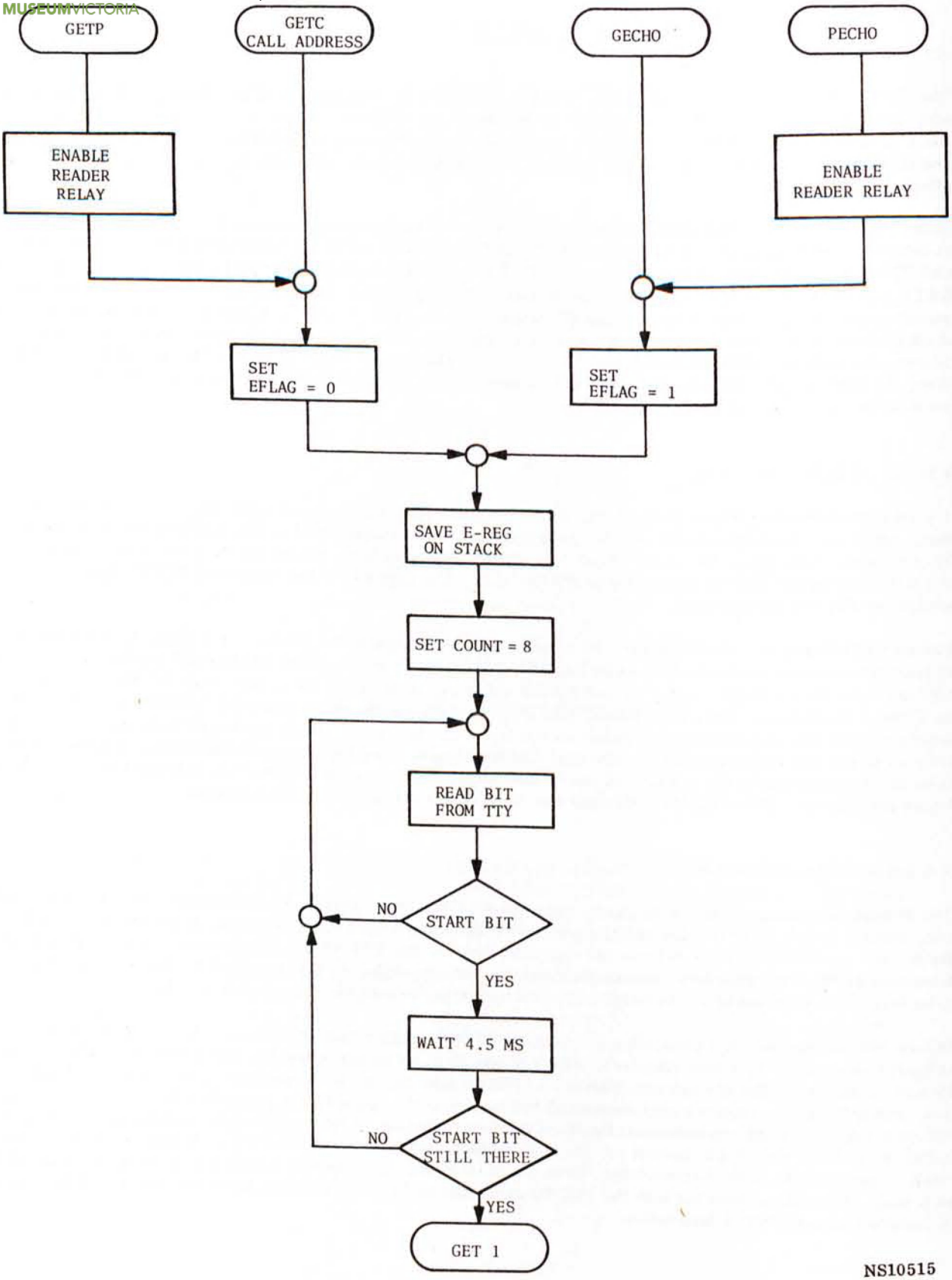


Figure 3-5. GETP, GETC, GECHO, and PECHO Subroutines (1 of 2)

3.6.1.5 PUTC Subroutine

The PUTC Subroutine (figure 3-6) transmits the value contained in the SC/MP accumulator to the TTY for printout as an ASCII character (refer to table 3-13). Execution of PUTC, therefore, causes the original contents of the SC/MP Status Register to be lost. The contents of the SC/MP Accumulator and the remaining SC/MP registers, except for P3, are not affected.

Before PUTC may be called initially, P2 must be set to the high-order address of a software stack that allows at least five memory locations for use by PUTC, P3 must be set to the PUTC call address 7AE1 or the PUTC exit address 7B14, and the desired ASCII value must be loaded into the SC/MP Accumulator. After these actions are accomplished, PUTC may be called via an XPPC 3 Instruction. Upon completion, PUTC will return to the next sequential instruction of the applications program with the original ASCII value reinstated in the SC/MP Accumulator, and the SC/MP P3 register set to the exit address PUTC. Thus, if the ASCII value is to be transmitted to the TTY again, PUTC may be recalled via an XPPC 3 Instruction. If a new ASCII value is to be transmitted to the TTY, however, the new value must first be loaded into the SC/MP Accumulator before PUTC is recalled via an XPPC 3 Instruction.

3.6.1.6 MESG Subroutine

The MESG Subroutine (figure 3-7) enables a string of ASCII characters to be read out of sequential memory and to be transmitted to the TTY for printout; the first character is read out of the memory address specified in the call, and the remaining characters are read out of memory in sequence until either a 00 data byte is encountered (indicating completion of the message) or a TTY key is pressed to terminate manually the message printout. During execution of the MESG Subroutine, the Status Register is used for various program control functions, and the original contents of the Status Register are lost. The contents of the SC/MP Accumulator and the remaining SC/MP registers, except Pointer Register 3 (P3), are not affected.

NOTE

The entire message must be located on a single "page" or "wraparound" will occur; refer to 3.4.1, TTY DEBUG Commands.

Before MESG may be called initially, P2 must be set to the high-order address of a software stack that allows at least 10 memory locations for use by MESG, and P3 must be set to the MESG call address 7B16 or the MESG exit address 7B49. Following these two actions, MESG may be called via XPPC 3 Instruction that is followed by two data bytes that contain the high-order and low-order digits of the memory address containing the first character of the message. If, for example, the first character of the message was located at memory address 4000, the calling sequence would be as follows:

Memory Address	Contents
XXXX	3F XPPC 3 INSTRUCTION
XXXX + 1	40
XXXX + 2	00

When MESG is called, it reads the first character out of the specified memory location and transmits the character to the TTY. Then MESG continues to read additional characters out of sequential memory locations and to transmit the characters to the TTY until it is terminated via a 00 data byte or by pressing a TTY key. Upon termination, MESG sets the SC/MP Pointer Register 3 (P3) to the MESG exit address 7B49 and increments the saved PC contents by 2. Thus, the memory-reference address associated with the call is skipped over and the MESG return is to the next sequential instruction of the applications program.

When additional messages are to be transmitted to the TTY for printout, MESG may be recalled after each return via an XPPC 3 Instruction followed by two data bytes that specify the memory address containing the first character of the new message.

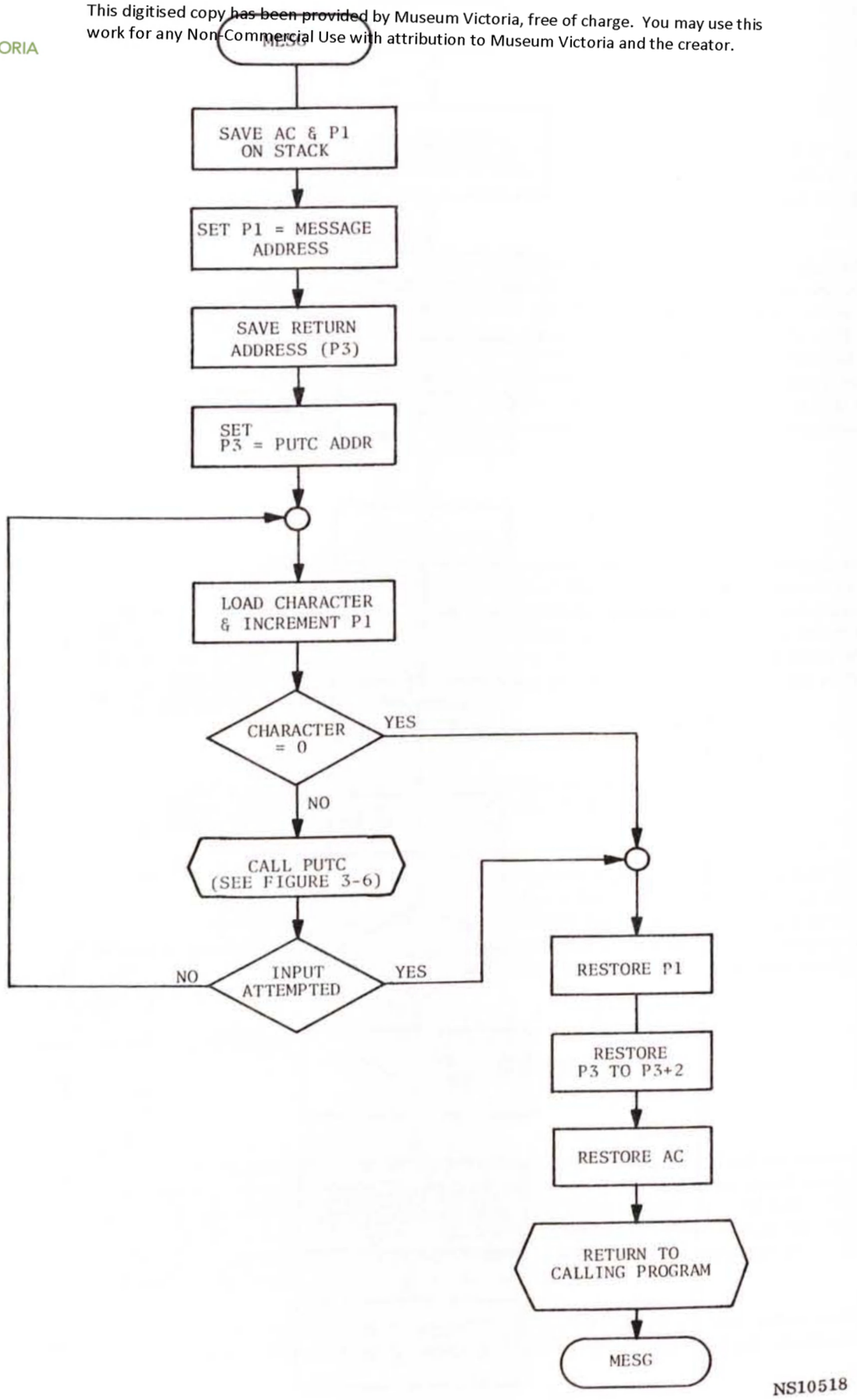
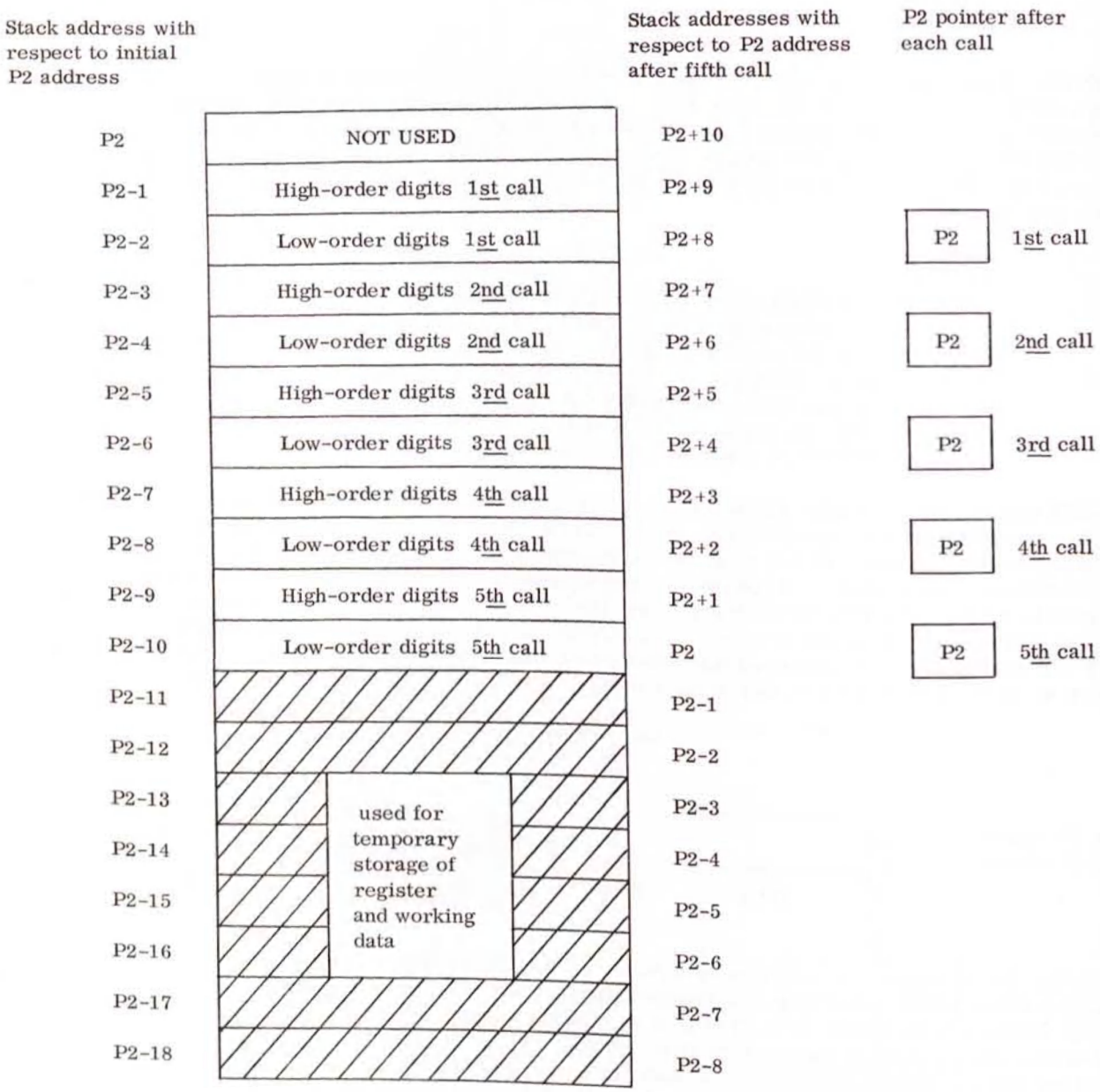
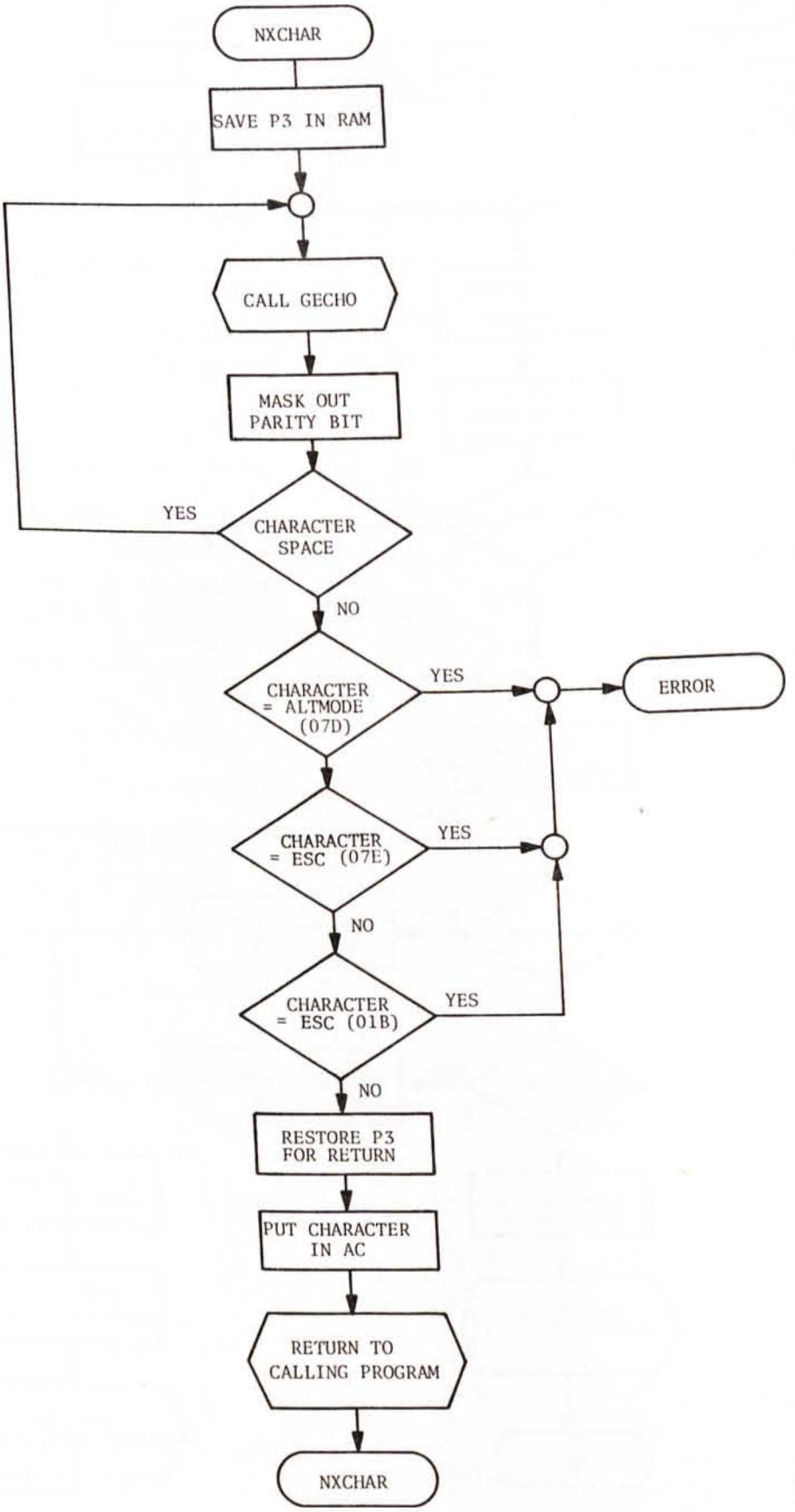


Figure 3-7. MESG Subroutine





NS10520

Figure 3-8. GHEX and GHEXE Subroutines (2 of 2)

3.6.1.9 PHEX Subroutine

The PHEX Subroutine (figure 3-9) transmits the value contained in the SC/MP Accumulator to the TTY for printout as a 2-digit hexadecimal value. Execution of PHEX, therefore, causes the original contents of the SC/MP Status Register to be lost. The contents of the SC/MP Accumulator and the remaining SC/MP registers, except for P3, are not affected.

Before PHEX may be called initially, P2 must be set to the high-order address of a software stack that allows at least 10 memory locations for use by PHEX, P3 must be set to the PHEX call address 7BB3, and the desired value must be loaded into the SC/MP Accumulator. After these actions are accomplished, PHEX may be called via an XPPC 3 Instruction. Upon completion, PHEX will return to the next sequential instruction of the applications program with the original value reinstated in the SC/MP Accumulator and the SC/MP P3 set to the exit address PHEXB. Thus, if PHEX is to be recalled to transmit another hexadecimal value to the TTY, P3 must be reloaded with the PHEX call address 7BB3; then, the new hexadecimal value must be loaded into the SC/MP Accumulator. Following these two actions, PHEX may be recalled via an XPPC 3 Instruction.

3.6.1.10 PHEXB Subroutine

The PHEXB Subroutine (figure 3-9) transmits the value contained in the SC/MP Accumulator to the TTY for printout as a 2-digit hexadecimal value followed by a trailing blank. Execution of PHEXB, therefore, causes the original contents of the SC/MP Status Register to be lost. The contents of the SC/MP Accumulator and the remaining SC/MP registers, except for P3, are not affected.

Before PHEXB may be called initially, P2 must be set to the high-order address of a software stack that allows at least 10 memory locations for use by PHEXB, P3 must be set to the PHEXB call address 7BAD or the PHEXB exit address 7BF3, and the desired value must be loaded into the SC/MP Accumulator. After these actions are accomplished, PHEXB may be called via an XPPC 3 Instruction. Upon completion, PHEXB will return to the next sequential instruction of the applications program with the original value reinstated in the SC/MP Accumulator and the SC/MP P3 set to the exit address PHEXB. Thus, if PHEXB is to be recalled to permit another 2-digit hexadecimal value to be printed out with trailing blank, the new hexadecimal value must be loaded into the SC/MP Accumulator before PHEXB is recalled via an XPPC 3 Instruction.

3.6.2 Digital Readout

The LCDS digital readout may be employed by an applications program for general display and/or trace purposes. Each digit of the readout is enabled via a unique address, and all of the digits share a common data input that controls lighting of the segments within the digits. The addresses of the digits and the data codes for hexadecimal values 0 through F are on the following page.

NOTE

The LCDS digital readout displays the hexadecimal values B and D in lower case (b and d) ~~because b and d are more readily distinguished~~ for easy distinction from the values 8 and 0.

Switch	Address	Data Output
0/READ PC	7010	BD0 = 1
1/READ P1	7010	BD1 = 1
2/READ P2	7010	BD2 = 1
3/READ P3	7010	BD3 = 1
4/READ AC	7008	BD0 = 1
5/READ EX	7008	BD1 = 1
6/READ ST	7008	BD2 = 1
7/READ MEM	7008	BD3 = 1
8/LOAD PC	7004	BD0 = 1
9/LOAD P1	7004	BD1 = 1
A/LOAD P2	7004	BD2 = 1
B/LOAD P3	7004	BD3 = 1
C/LOAD AC	7002	BD0 = 1
D/LOAD EX	7002	BD1 = 1
E/LOAD ST	7002	BD2 = 1
F/LOAD MEM	7002	BD3 = 1

Reading a particular row of pushbuttons is effected via a Load (LD) Instruction. Upon completion of the Load Instruction the contents of the Accumulator will either be zero (indicating that none of the pushbuttons in the row was pressed) or greater than zero (indicating that one of the pushbuttons in the row was pressed). If the contents of the Accumulator are greater than zero, the position of the high bit will indicate the pushbutton that was pressed.

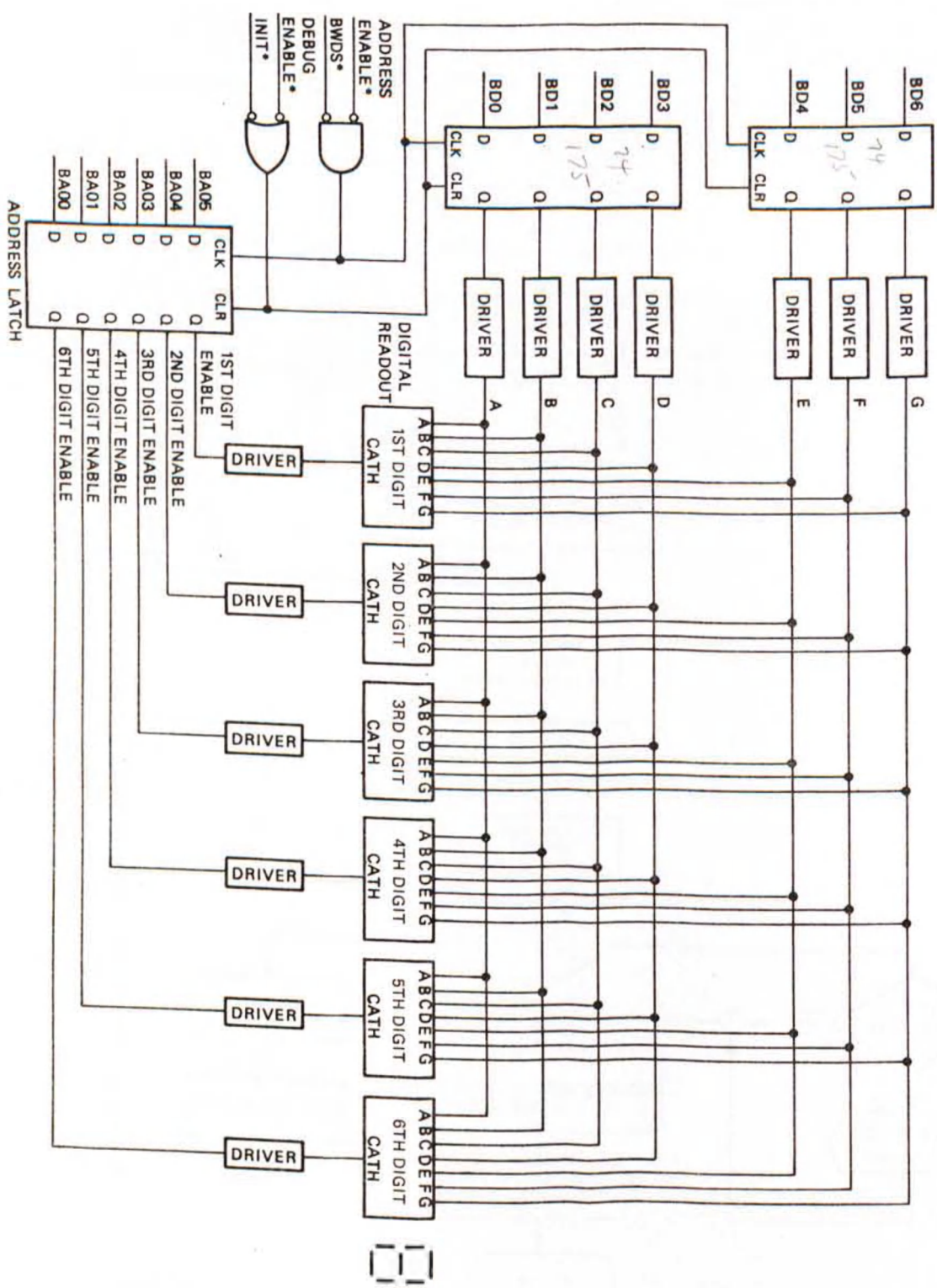


Figure 3-10. LCDS Digital Readout and Control Circuit

NS10718

A general summary of the register assignments and the call/return sequence for the GETP, GETC, GECHO and PECHO subroutines follows.

a. Register Assignment Prior to Call:

P2 — high order address of software stack that contains at least five memory locations

P3 — GETP 7A84
GETC 7A88
PECHO 7A8C
GECHO 7A90

b. Subroutine Call: XPPC 3 Instruction

c. Subroutine Return: to XPPC 3 Instruction address + 1

d. Register Assignment Upon Return:

P1 — unchanged
P2 — high-order address of stack
P3 — GECHO repeat address 7ADF
AC — ASCII input character
EX — unchanged

	CY/L	OV	S_B	S_A	IE	F2	F1	F0
S -	?	U	I	I	U	U	0	0
Bit	7	6	5	4	3	2	1	0

where

U - unchanged
I - read-only bit (logic state is controlled by external input)
? - CY/L flag reflects state of last TTY output data bit

4.3.2 PUTC Subroutine

The PUTC subroutine transmits the value contained in the SC/MP Accumulator to the TTY for printout. Overall execution of the PUTC subroutine is shown in figure 4-3, and an annotated program listing is provided in figure 4-4. For detailed information on how this subroutine may be incorporated into an applications program, refer to 3.6.1.5.

A general summary of the register assignments and the call/return sequence for the PUTC subroutine follows.

a. Register Assignment Prior to Call:

P2 — high-order address of software stack that contains at least five memory locations
P3 — PUTC call address 7AE1
AC — ASCII output character

b. Subroutine Call: XPPC 3 Instruction

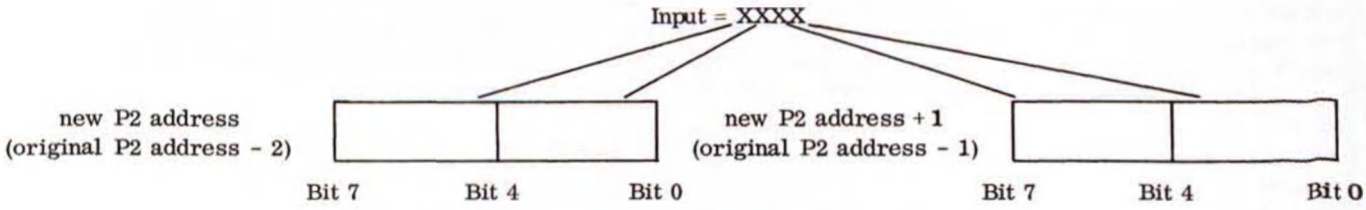
c. Subroutine Return: to XPPC 3 Instruction address + 1

d. Register Assignment Upon Return:

P1 — unchanged
P2 — high-order address of stack
P3 — PUTC repeat address 7ADF
AC — unchanged (ASCII output value)
EX — unchanged

4.3.4 GHEX and GHEXE Subroutines

The GHEX and GHEXE subroutines provide an ASCII-to-hexadecimal conversion function that enables ASCII data to be read in and stored on a software stack in 4-digit hexadecimal format. The only difference in execution of the two subroutines is that the GHEXE subroutine reads in the first ASCII character from the Extension Register and subsequent ASCII characters from the TTY keyboard, whereas the GHEX subroutine reads in all of the ASCII characters from the TTY keyboard. During execution of either subroutine, reading in of ASCII data continues until escape character ESC (data code 7E or 1B) or ALTMODE (data code 7D) is detected, or until an ASCII terminator is received (any ASCII character except 0-9, A-F, ESC, and ALT MODE). When an escape character is detected, an error message is first printed out ((CR) (LF) , ?), and then a branch is made to a DEBUG firmware command loop to terminate subroutine execution and inhibit a return to the calling program. When an ASCII terminator is detected, it causes the subroutine to return to the calling program with the P2 register decremented by two and with a 4-digit hexadecimal value stored on the stack as shown below. (If less than four digits are read in before the ASCII terminator is received, the missing digits are treated as leading zeroes; if more than four digits are read in, only the last four digits are saved.)



Overall execution of the GHEX and GHEXE subroutines is shown in flowchart form in figure 4-7, an annotated program listing is provided in figure 4-8, and supplementary descriptions of the more involved subroutine operations are provided in figures 4-9 through 4-11. For detailed information on how the GHEX and GHEXE subroutines may be incorporated into an applications program, refer to 3.6.1.7 and 3.6.1.8.

A general summary of the register assignments and the call/return sequence for the GHEX and GHEXE subroutines follows:

a. Register Assignment Prior to Call:

P2 — high order address of software stack that contains at least 11 memory locations
P3 — GHEXE call address 7B4B or
GHEX call address 7B4F
EX — first ASCII character (GHEXE subroutine call only)

b. Subroutine Call: XPPC 3 Instruction

c. Subroutine Return: to XPPC 3 Instruction address + 1

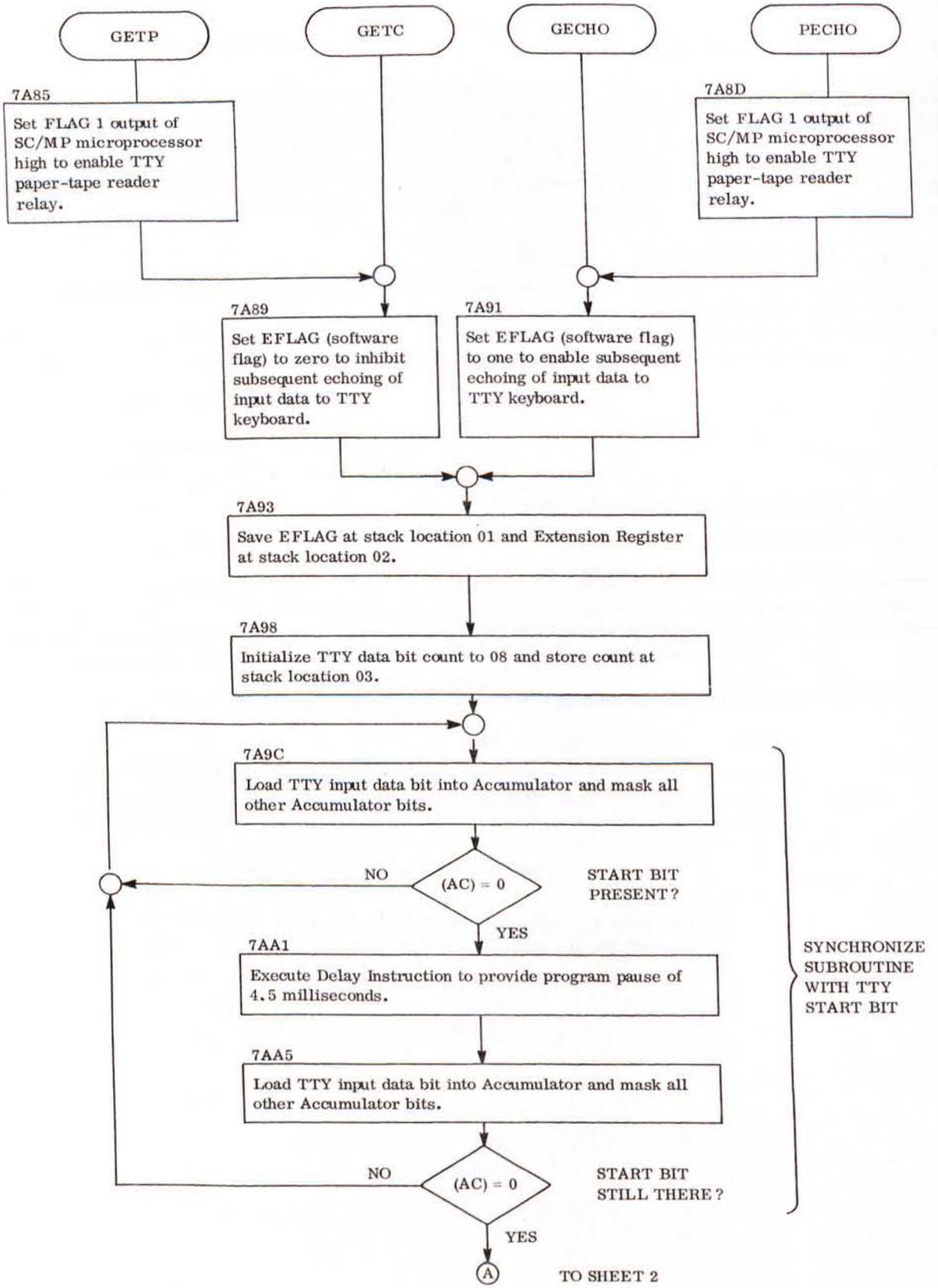
d. Register Assignment Upon Return:

P1 — unchanged
P2 — high order address of stack - 2
P3 — GHEX repeat address 7BAB
AC — ASCII Terminator
EX — ASCII Terminator

	CY/L	OV	S_B	S_A	IE	F2	F1	F0
S -	?	?	I	I	U	U	0	0
Bit	7	6	5	4	3	2	1	0

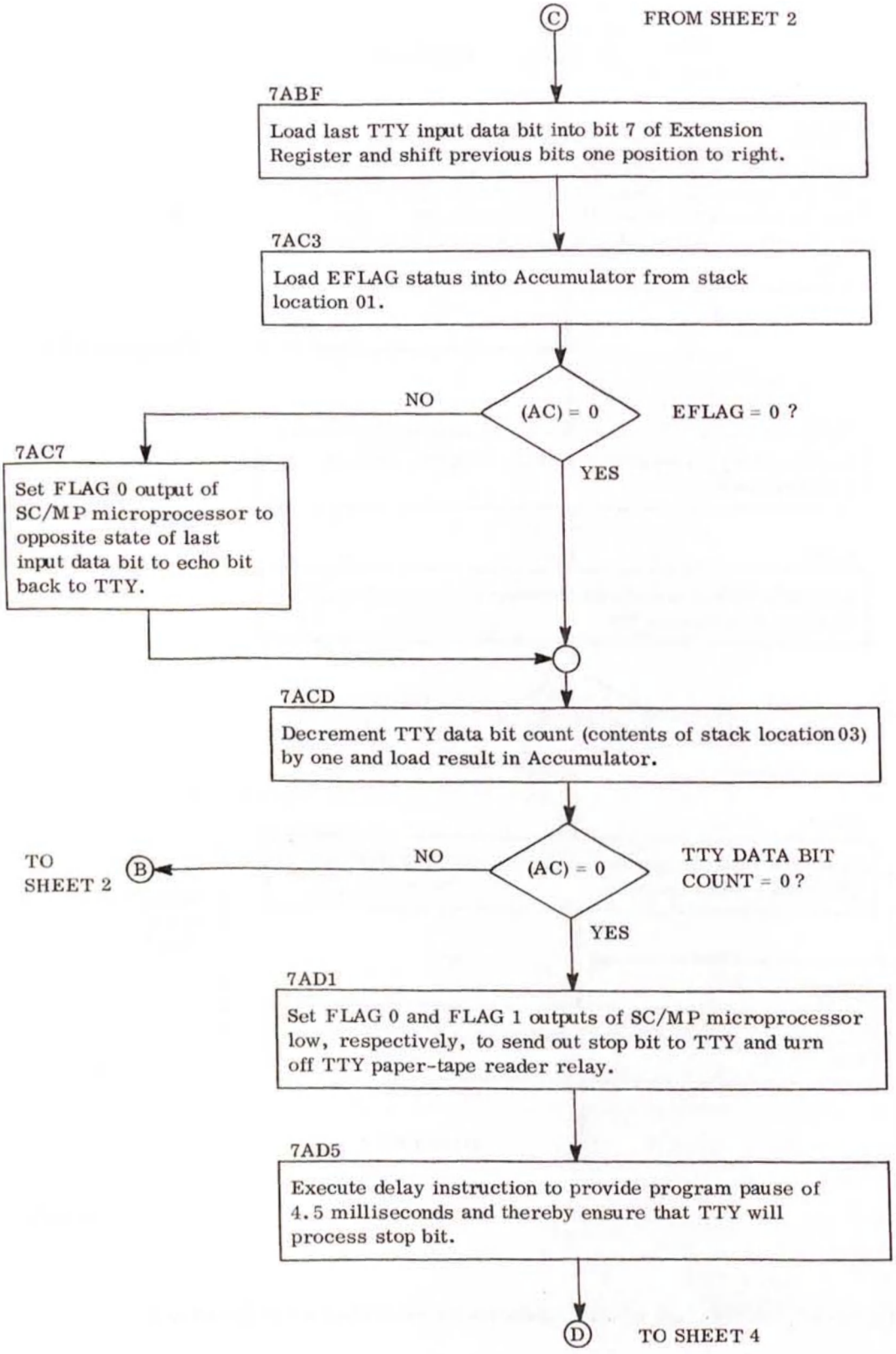
where

U - unchanged
I - read-only bit (logic state is controlled by external input)
? - unknown (varies according to ASCII input value)



NS10644

Figure 4-1. GETP, GETC, GECHO, and PECHO Subroutine Detailed Flowchart (Sheet 1 of 4)



NS10646

Figure 4-1. GETP, GETC, GECHO, and PECHO Subroutine Detailed Flowchart (Sheet 3 of 4)

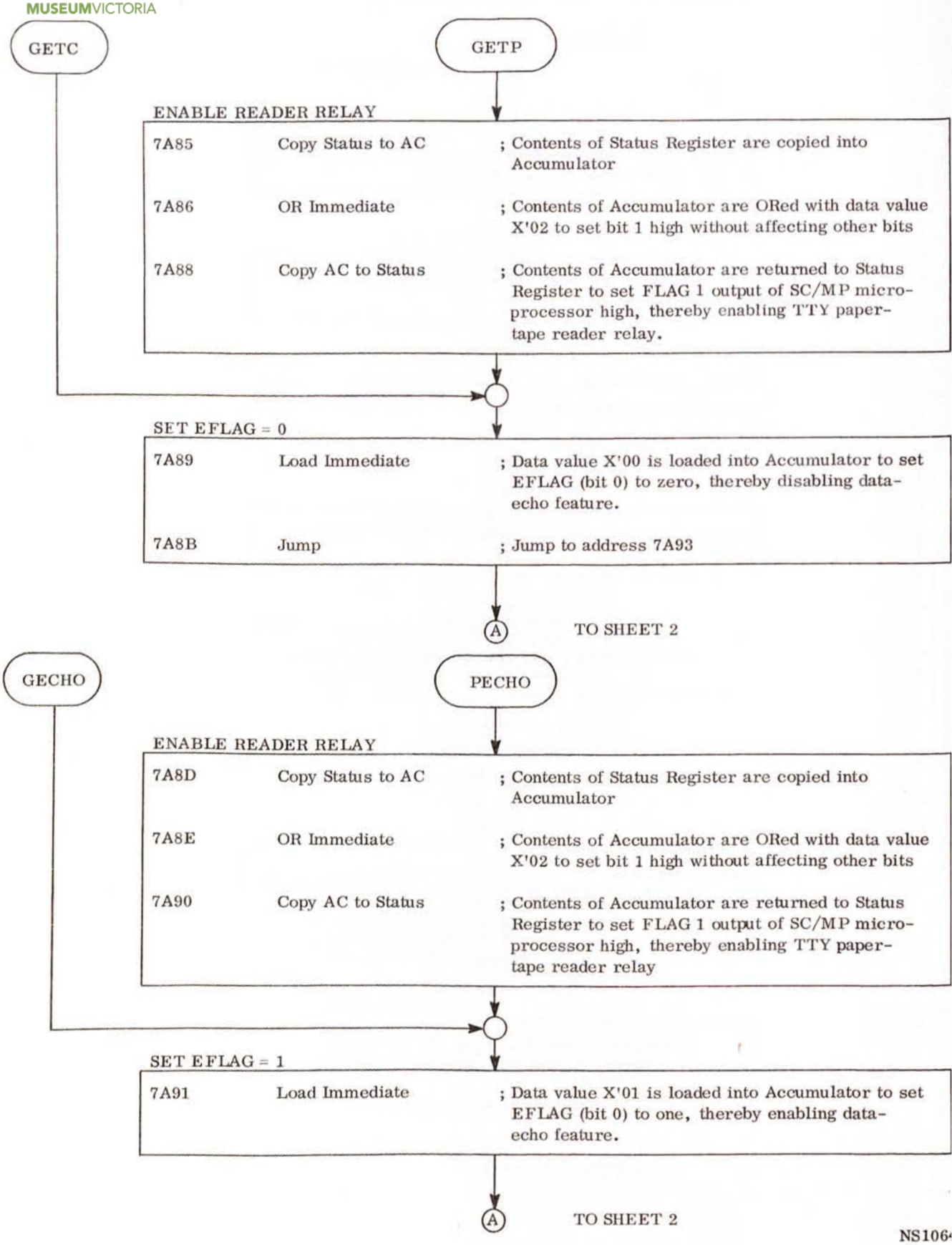


Figure 4-2. GETP, GETC, GECHO, and PECHO Subroutine — Annotated Instruction Listing (Sheet 1 of 8)

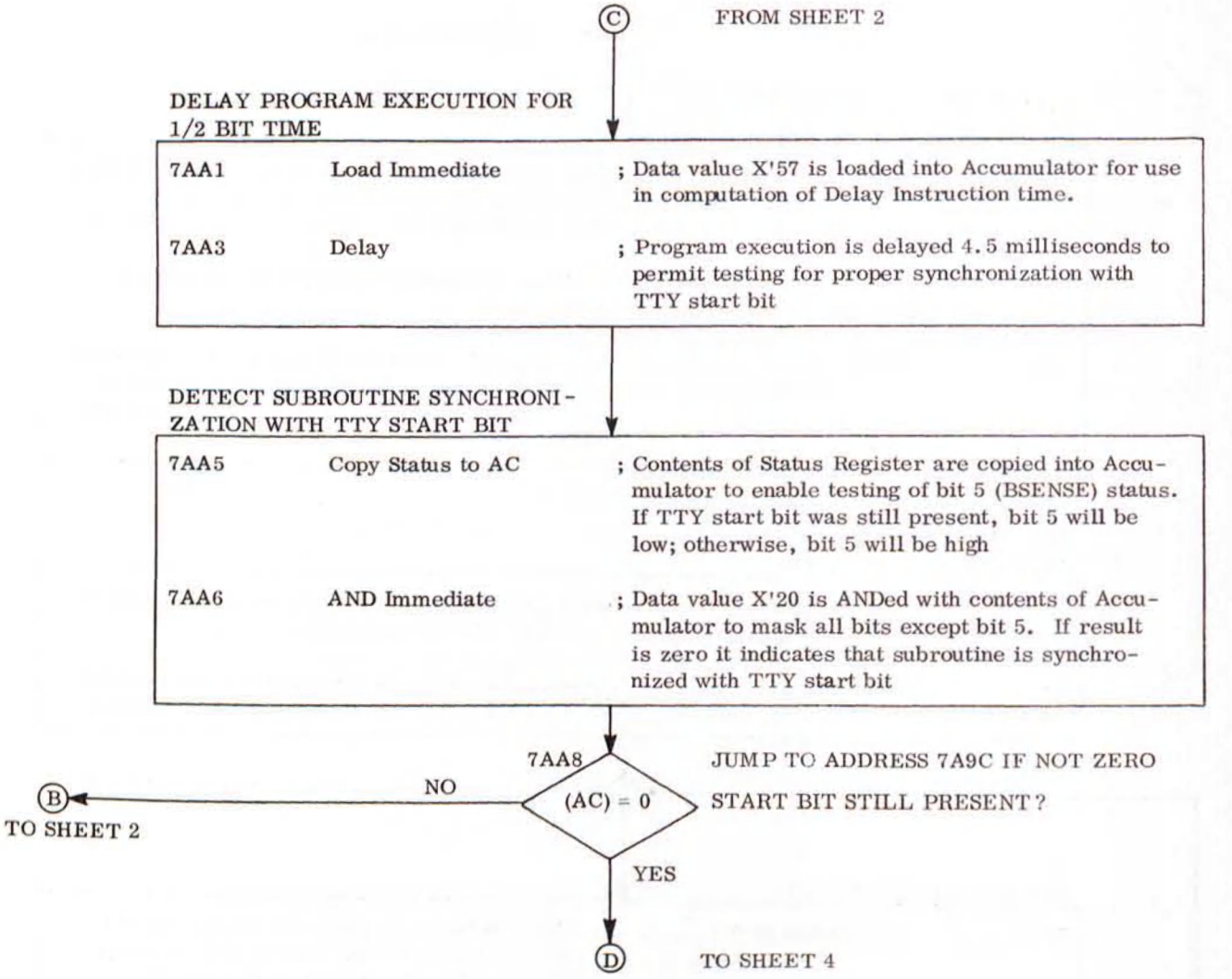


Figure 4-2. GETP, GETC, GECHO, and PECHO Subroutine — Annotated Instruction Listing (Sheet 3 of 8)

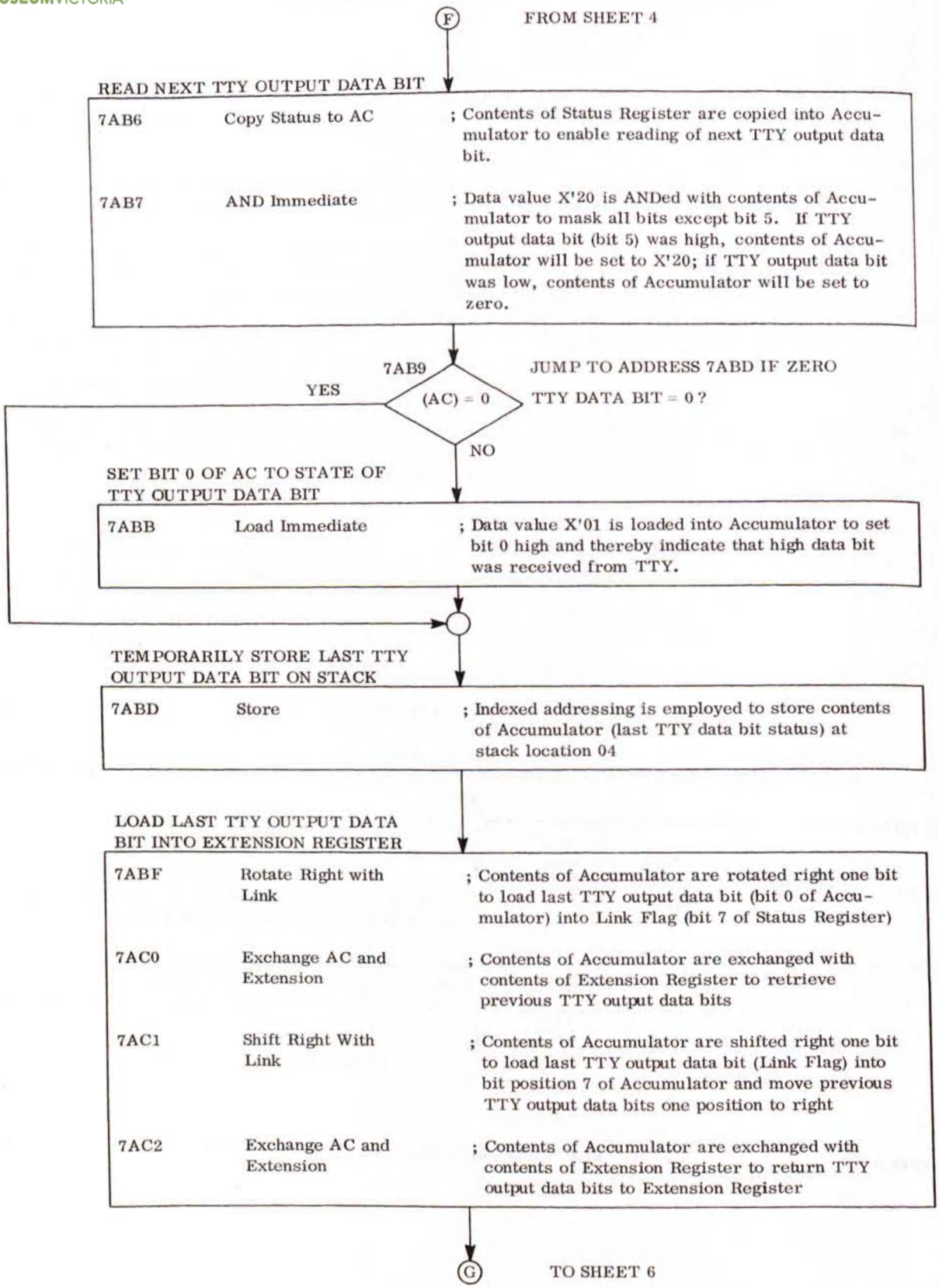


Figure 4-2. GETP, GETC, GECHO, and PECHO Subroutine — Annotated Instruction Listing (Sheet 5 of 8)

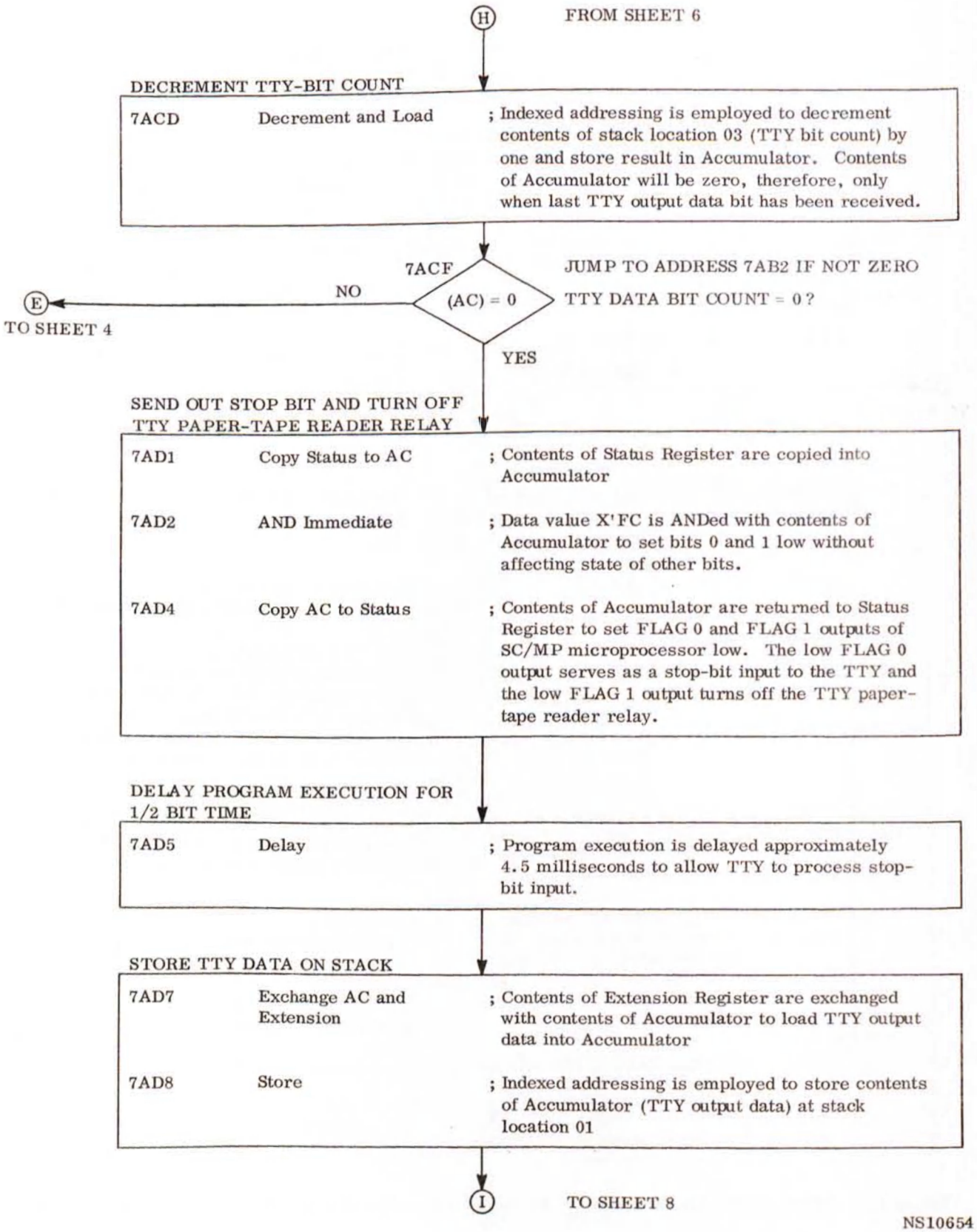
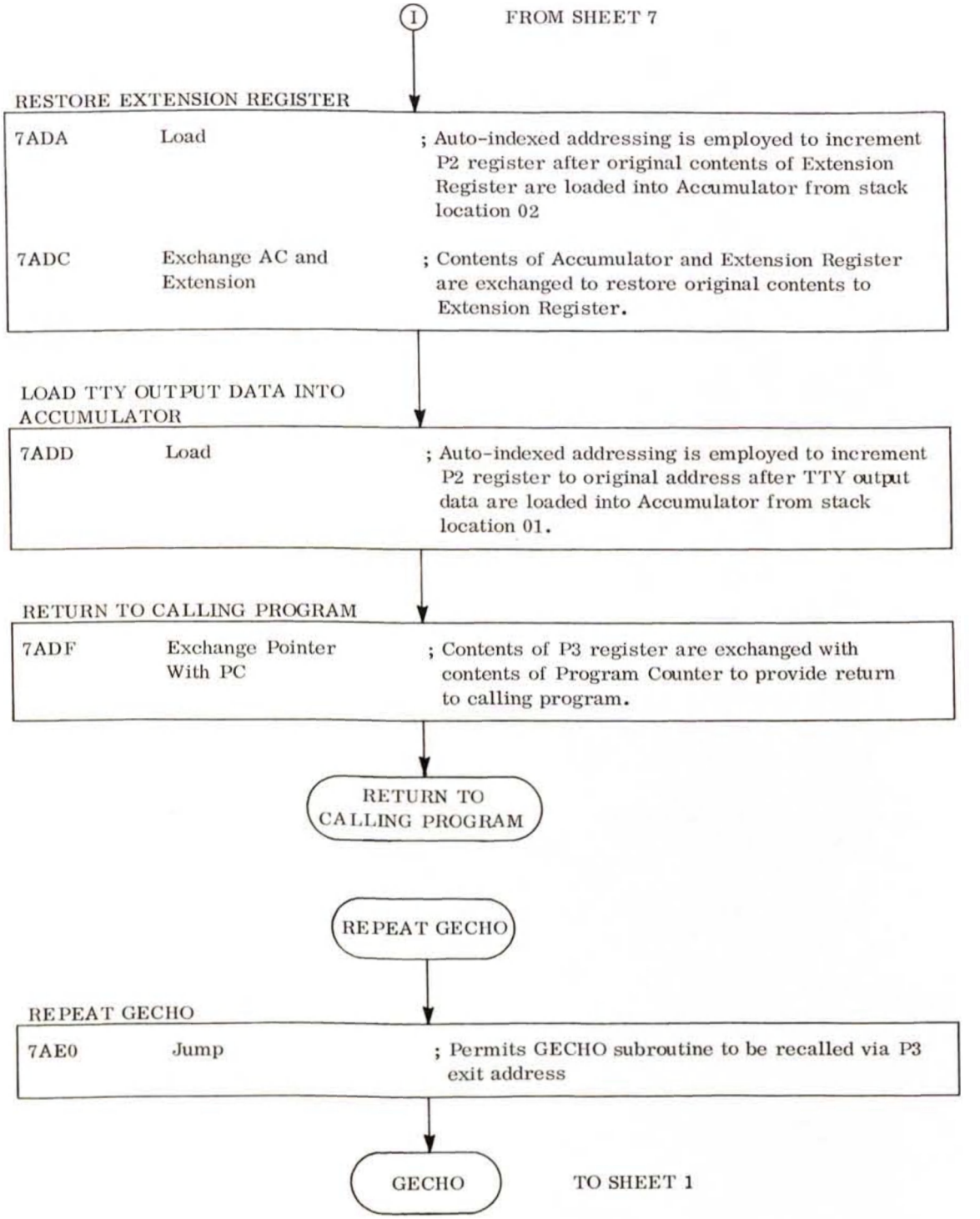


Figure 4-2. GETP, GETC, GECHO, and PECHO Subroutine — Annotated Instruction Listing (Sheet 7 of 8)



NS10655

Figure 4-2. GETP, GETC, GECHO, and PECHO Subroutine — Annotated Instruction Listing (Sheet 8 of 8)

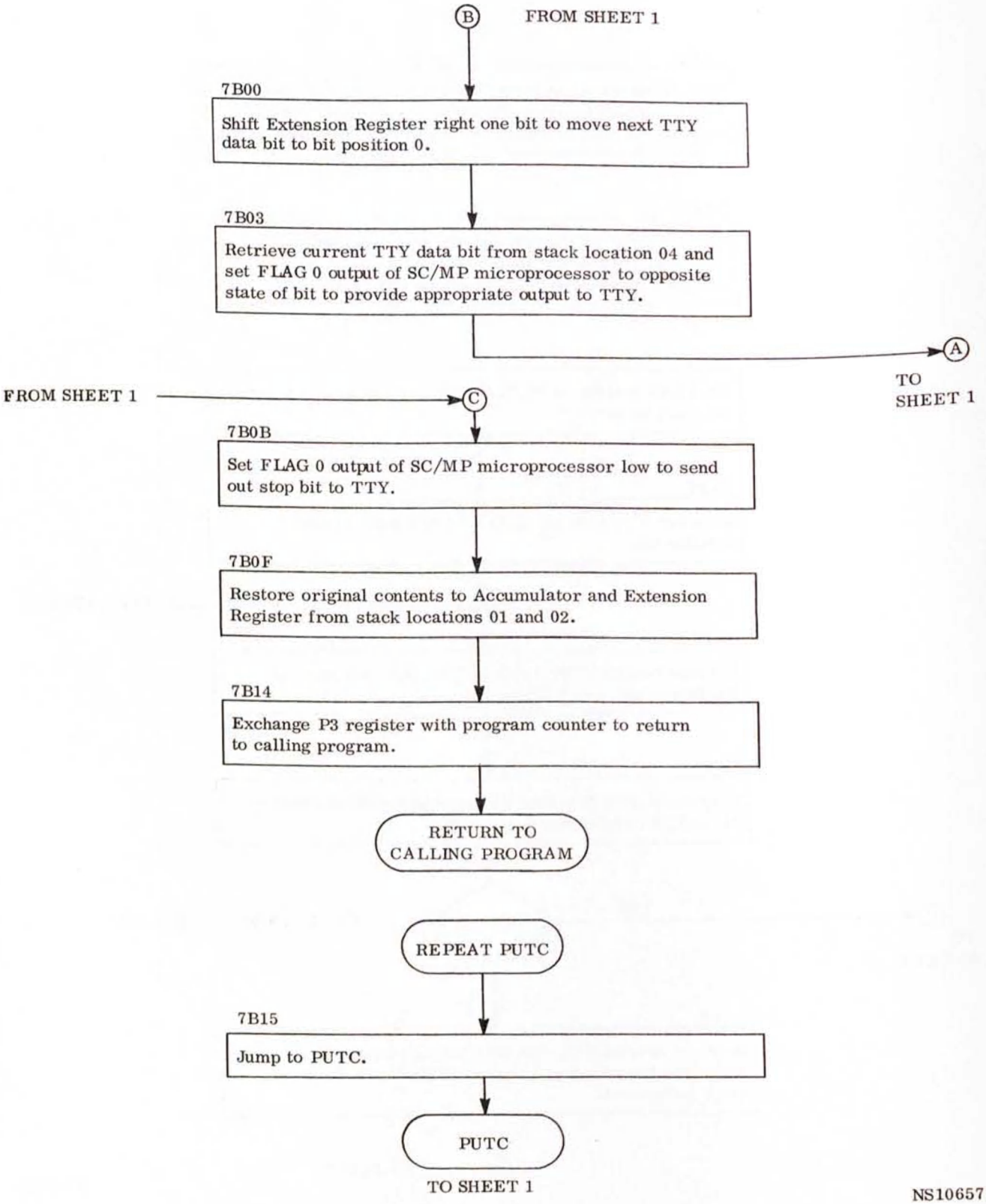


Figure 4-3. PUTC Subroutine — Detailed Flowchart (Sheet 2 of 2)

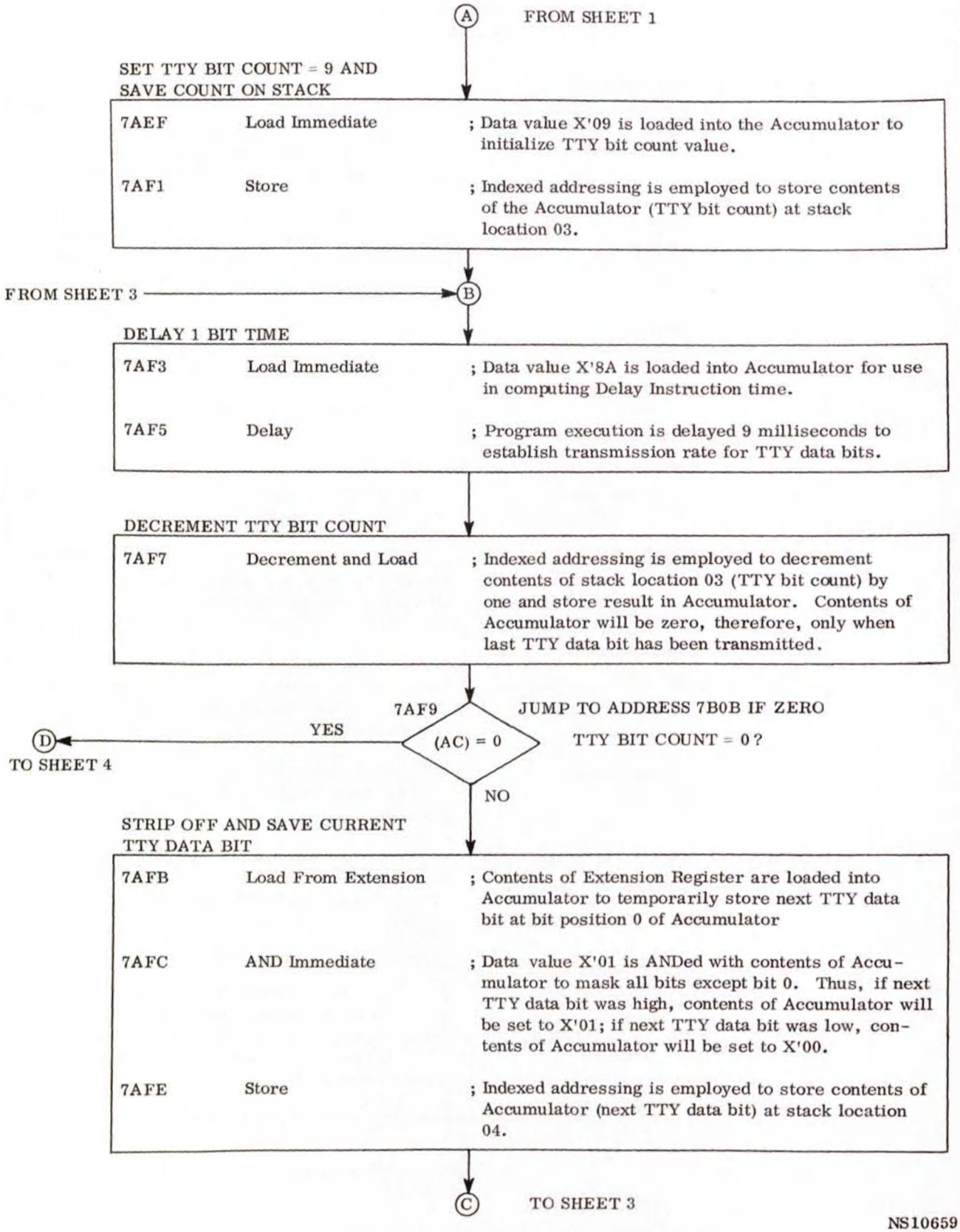


Figure 4-4. PUTC Subroutine — Annotated Instruction Listing (Sheet 2 of 4)

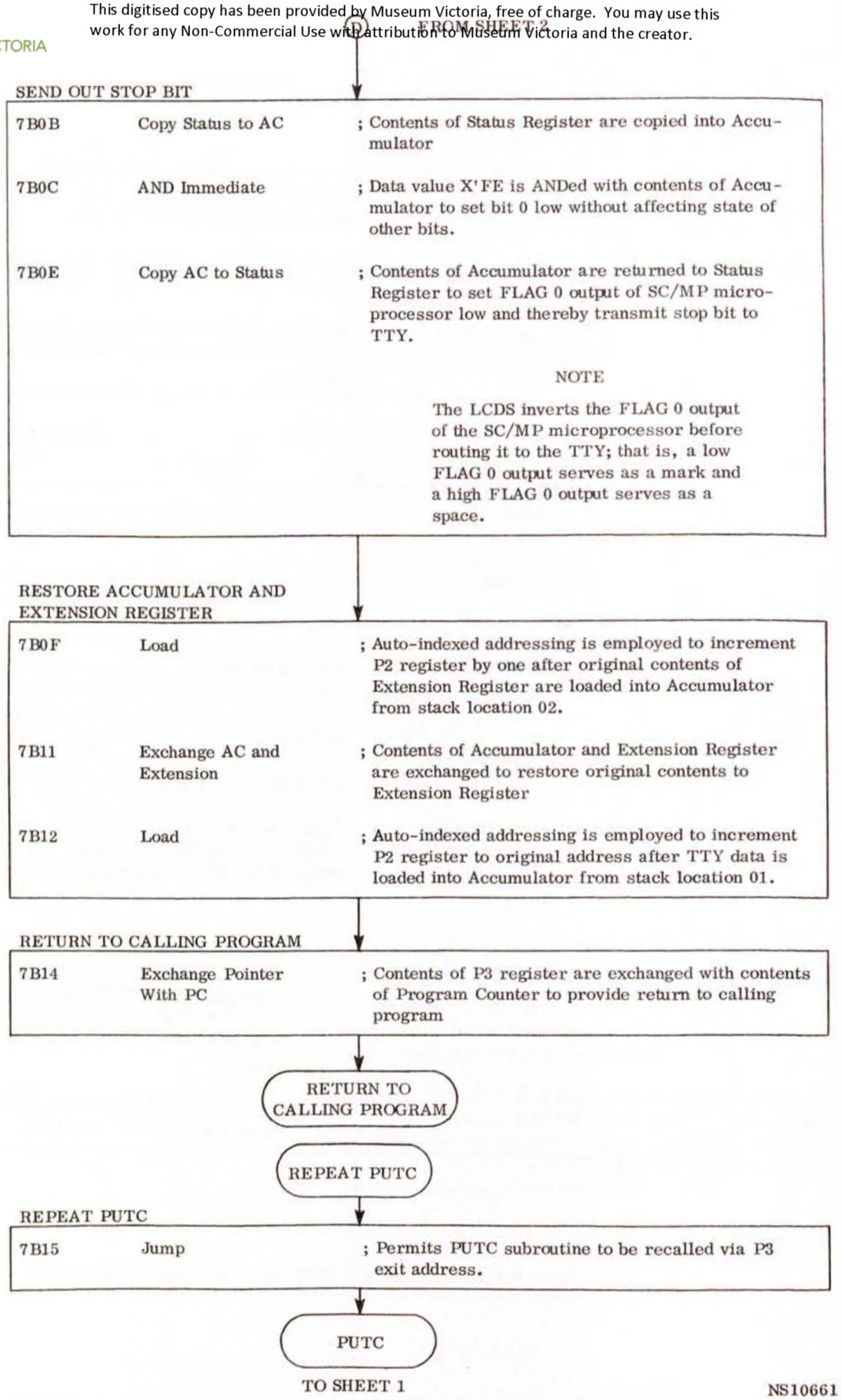


Figure 4-4. PUTC Subroutine — Annotated Instruction Listing (Sheet 4 of 4)

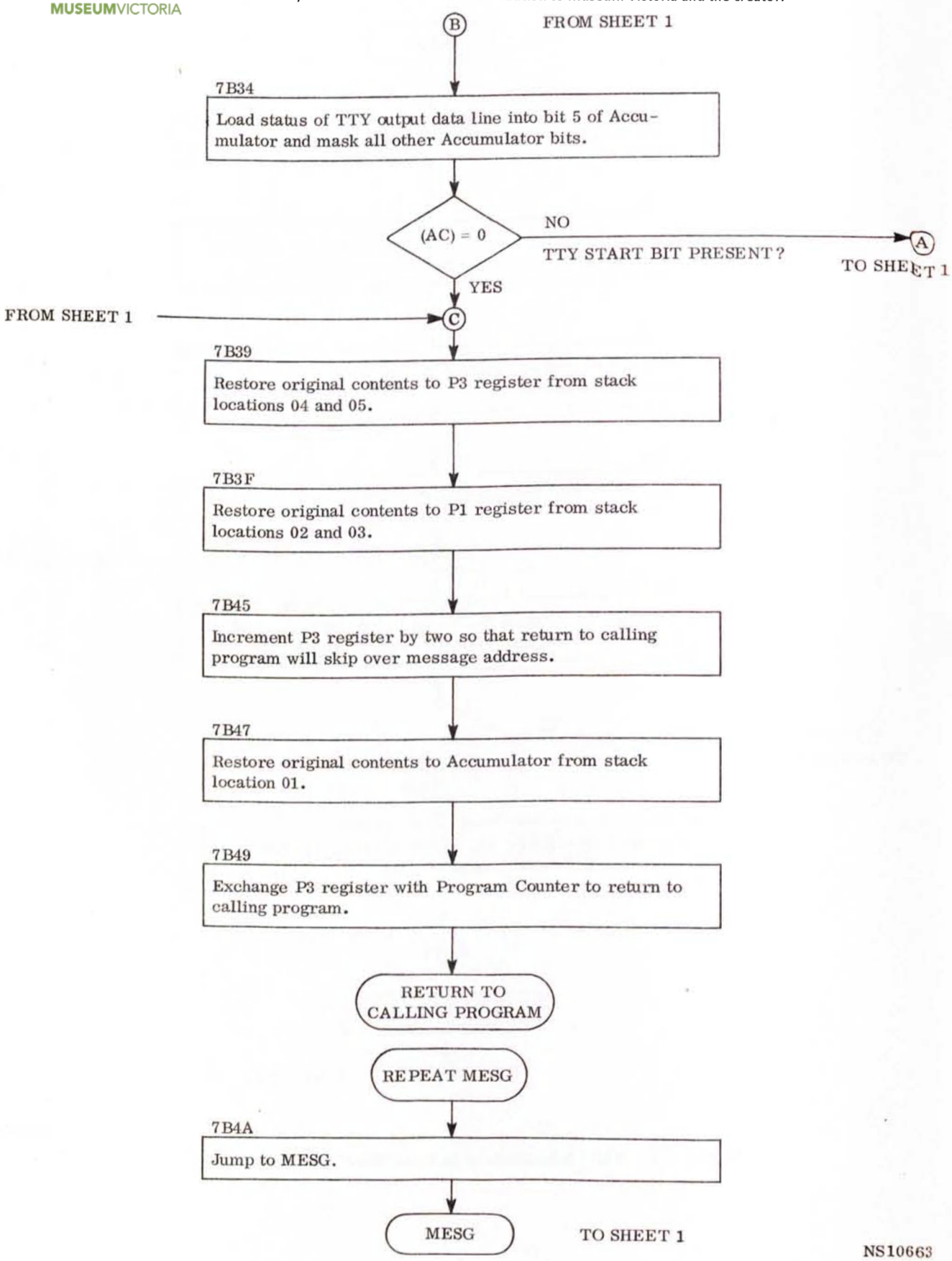


Figure 4-5. MESG Subroutine — Detailed Flowchart (Sheet 2 of 2)

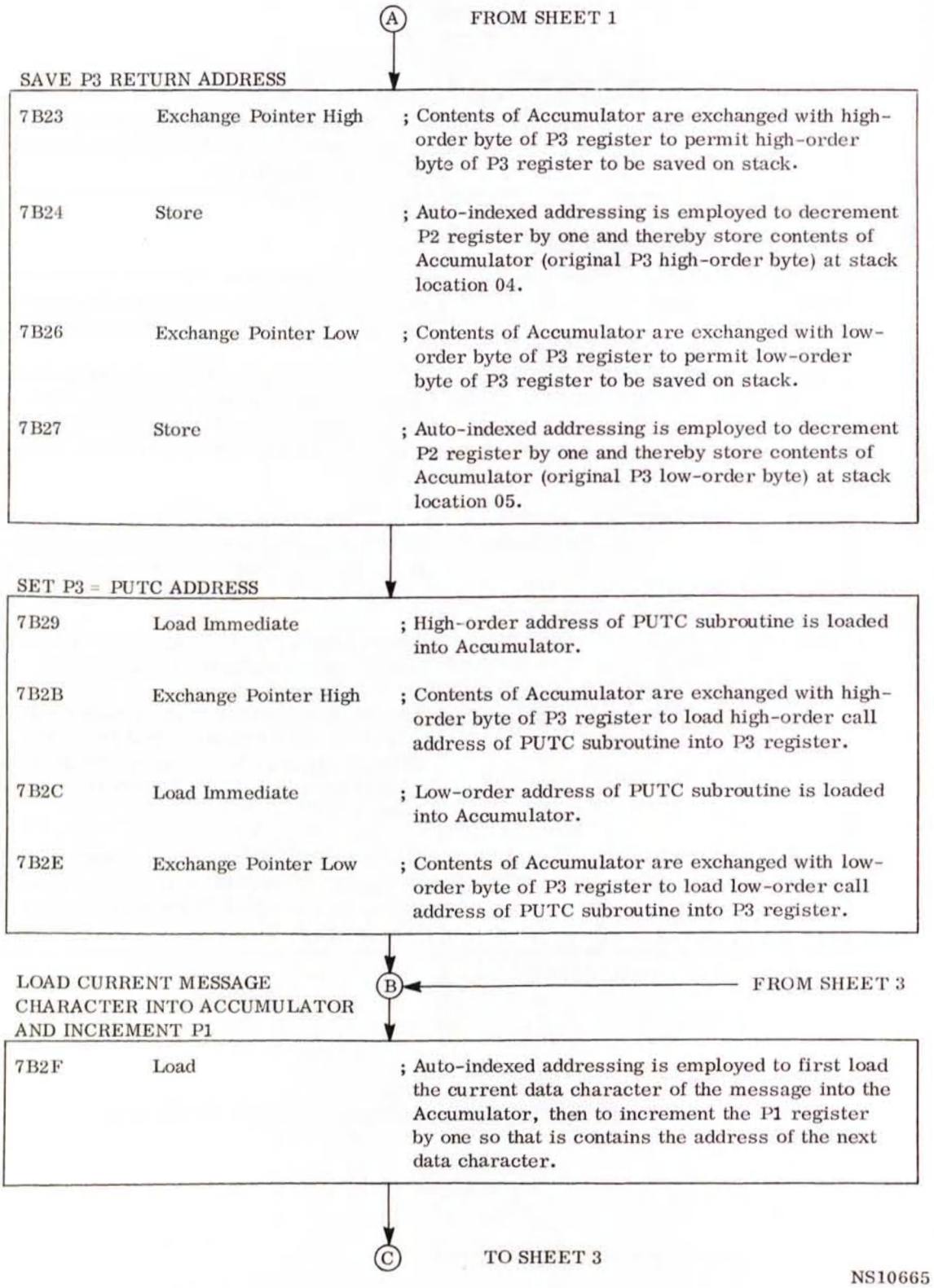


Figure 4-6. MESG Subroutine — Annotated Instruction Listing (Sheet 2 of 5)

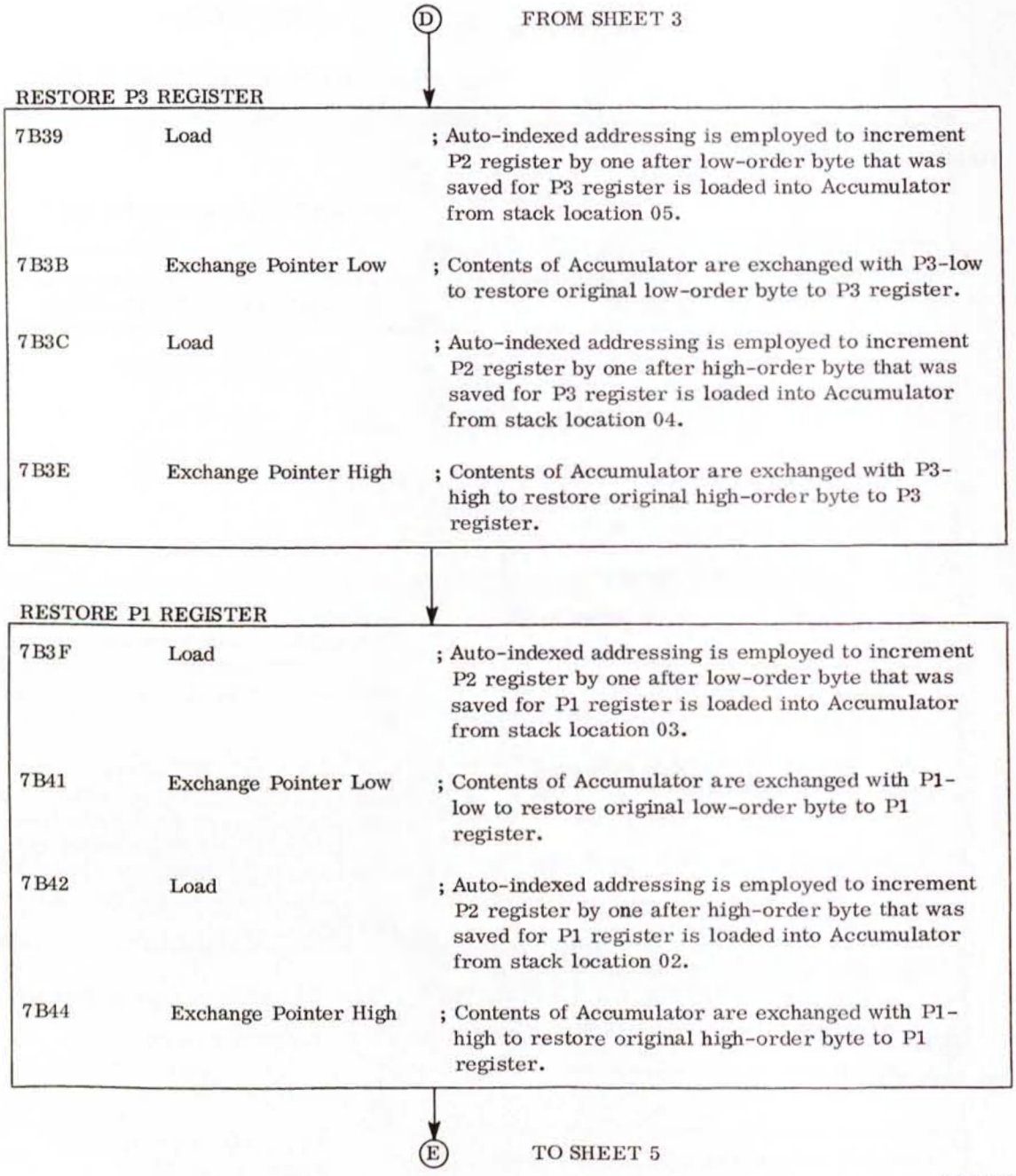


Figure 4-6. MESG Subroutine — Annotated Instruction Listing (Sheet 4 of 5)

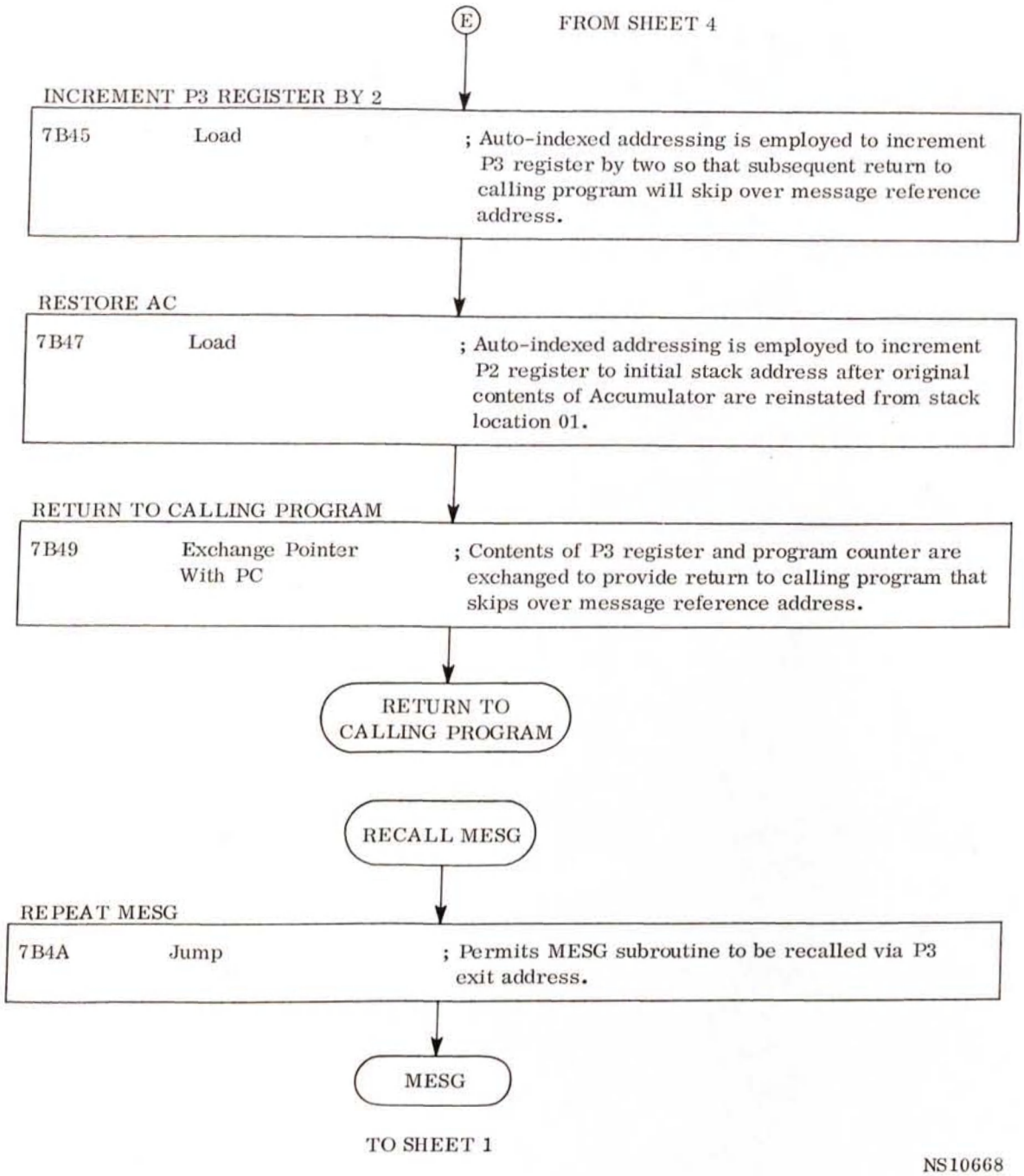
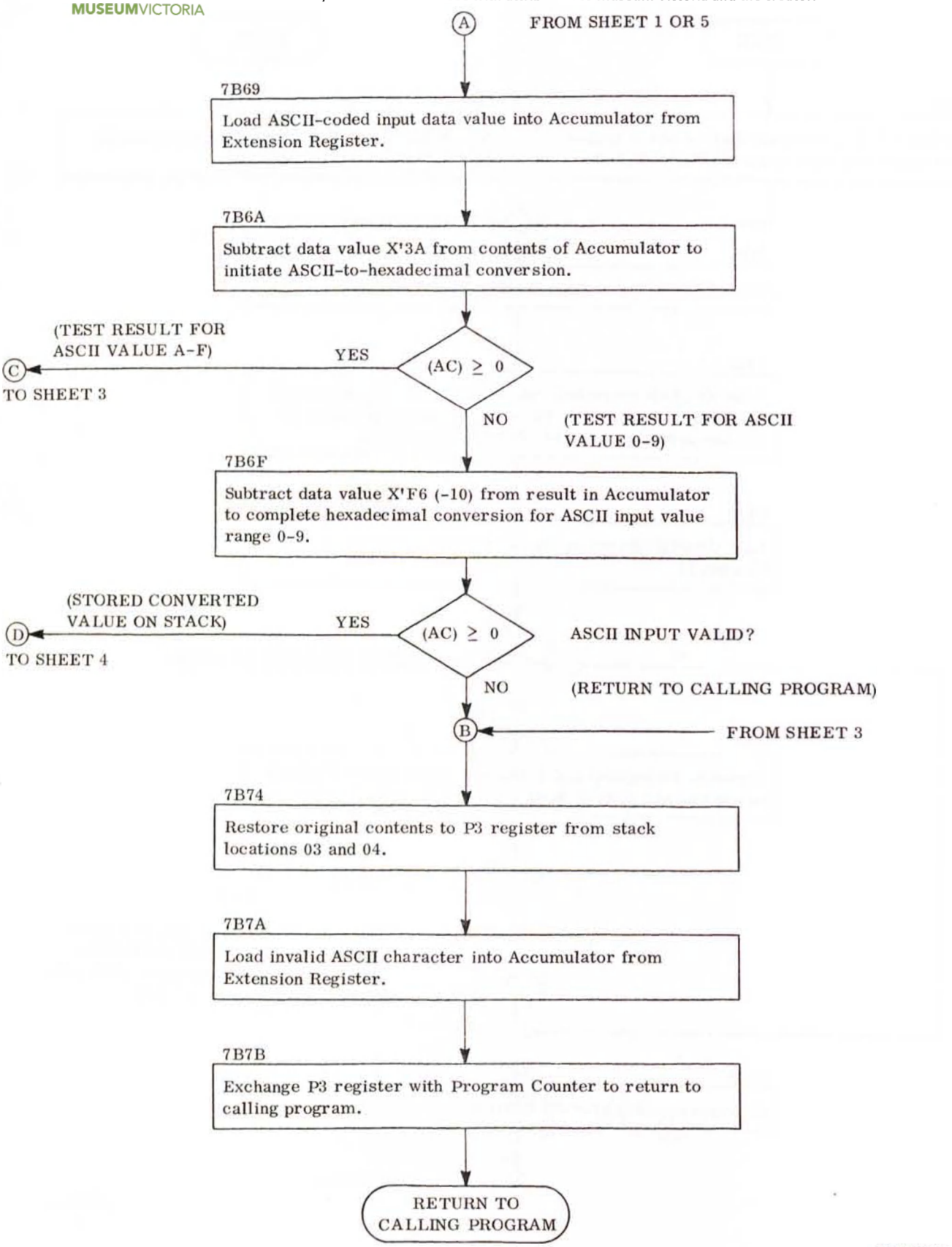
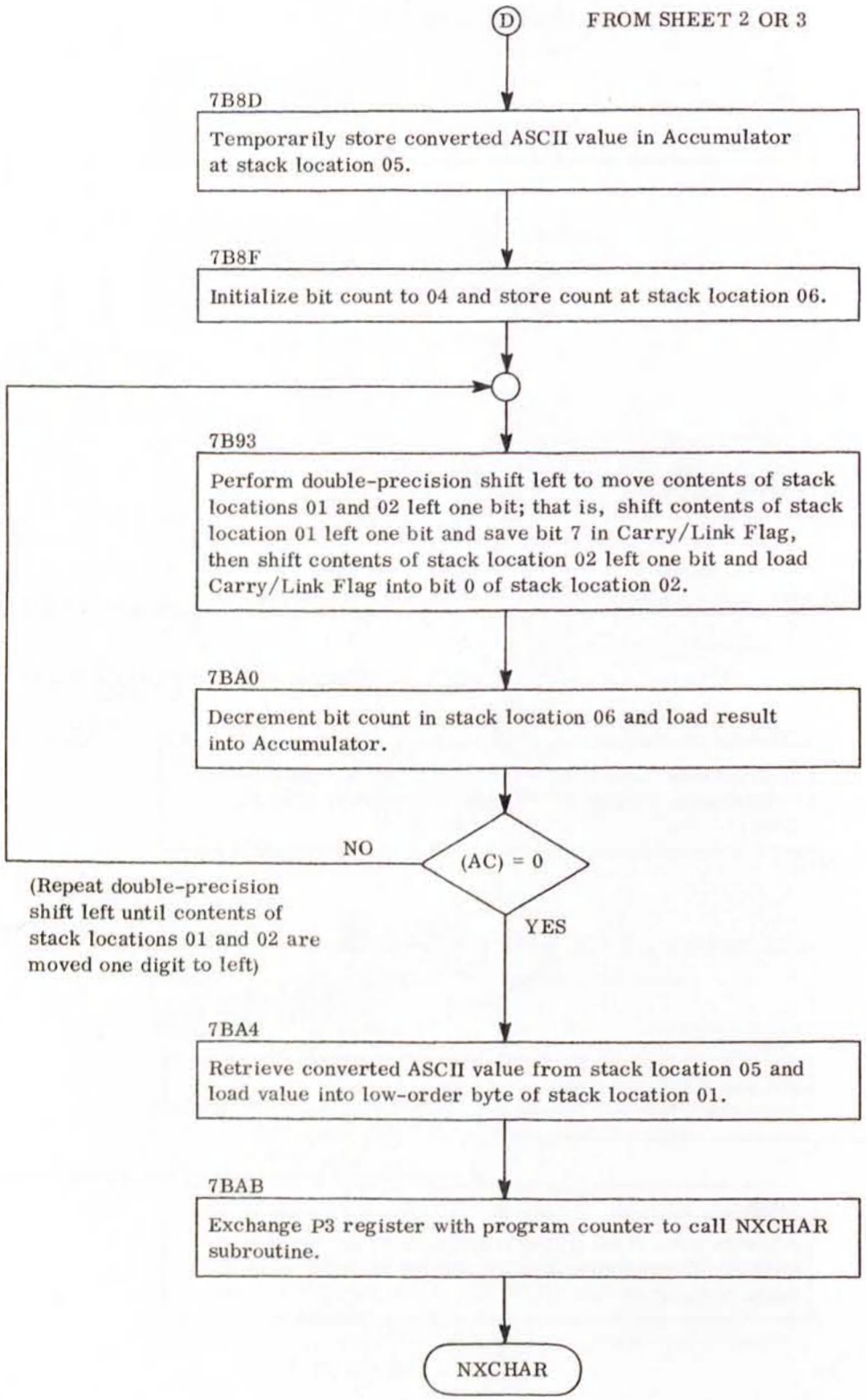


Figure 4-6. MESG Subroutine — Annotated Instruction Listing (Sheet 5 of 5)



NS10670

Figure 4-7. GHEX and GHEXE Subroutine — Detailed Flowchart (Sheet 2 of 7)



NS10672

Figure 4-7. GHEX and GHEXE Subroutine — Detailed Flowchart (Sheet 4 of 7)

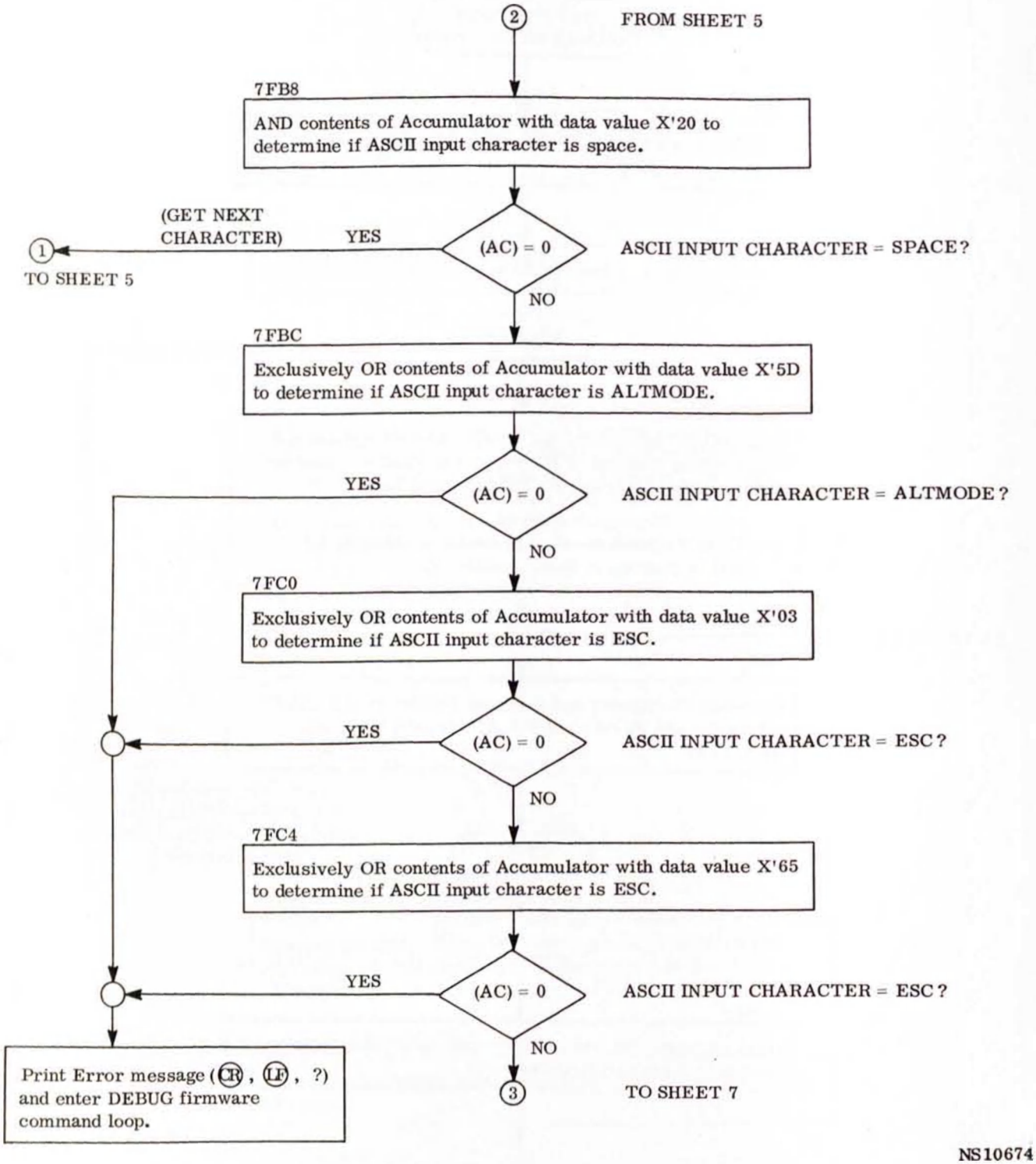


Figure 4-7. GHEX and GHEXE Subroutine — Detailed Flowchart (Sheet 6 of 7)

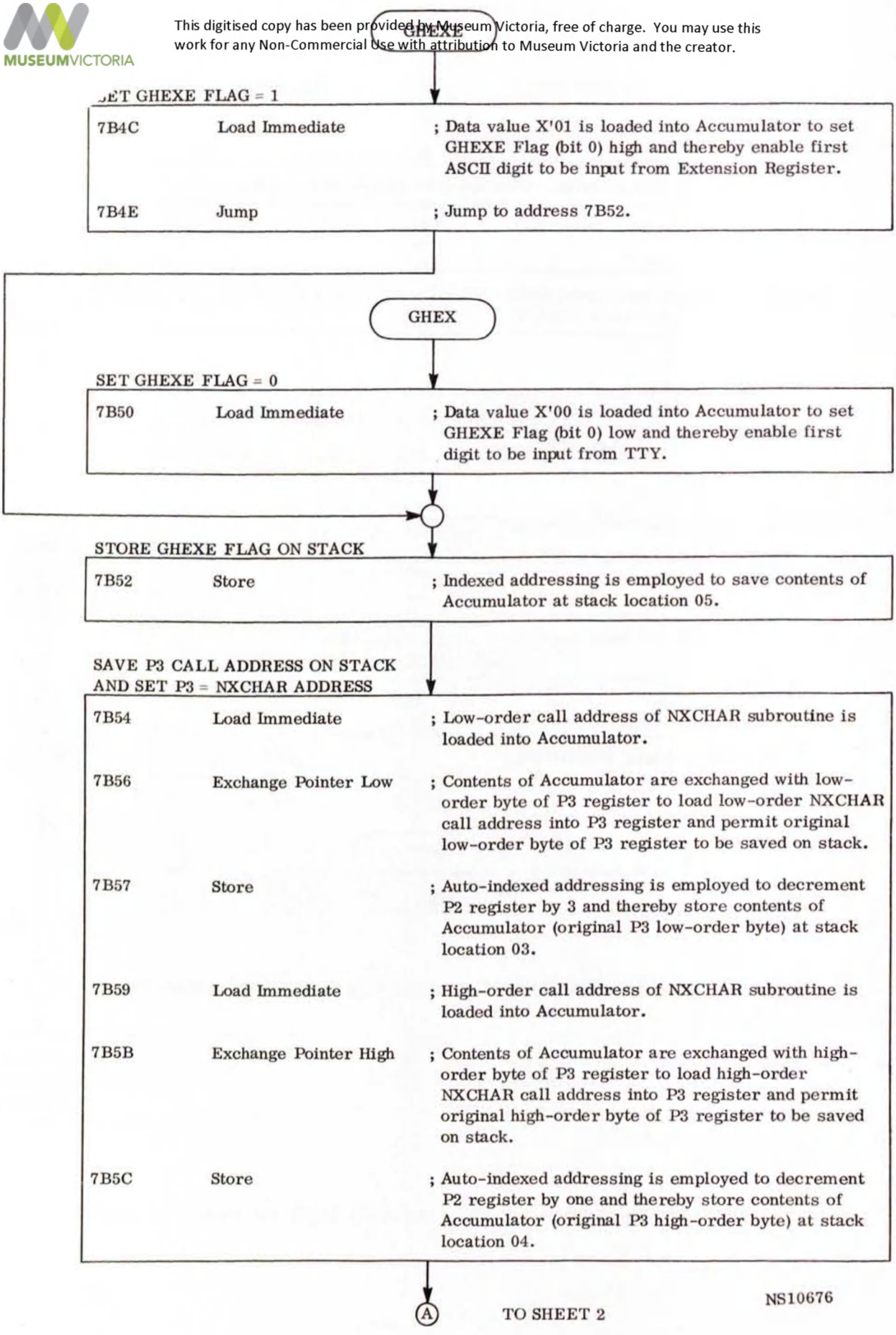


Figure 4-8. GHEX and GHEXE Subroutine — Annotated Instruction Listing (Sheet 1 of 12)

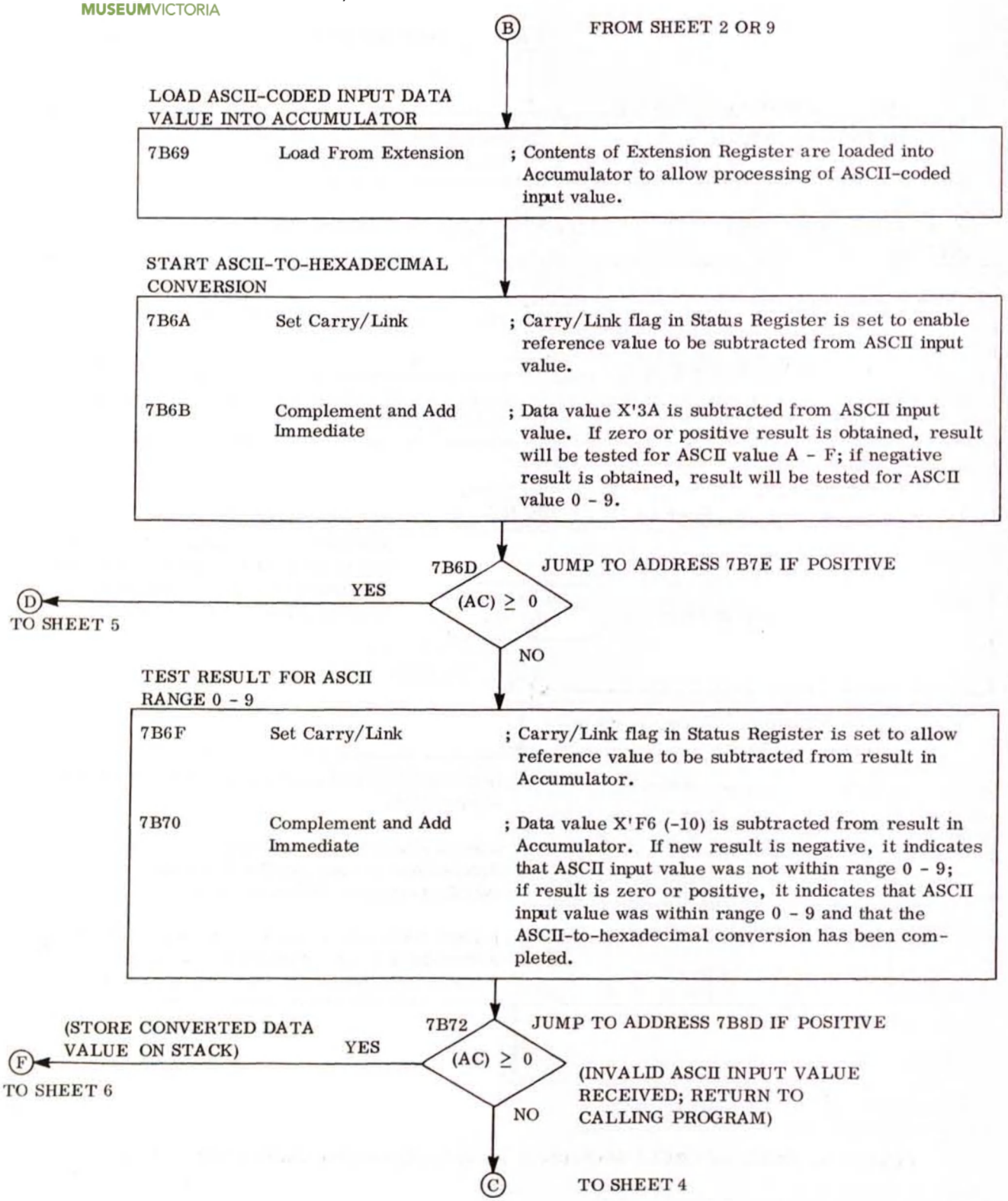


Figure 4-8. GHEX and GHEXE Subroutine — Annotated Instruction Listing (Sheet 3 of 12)

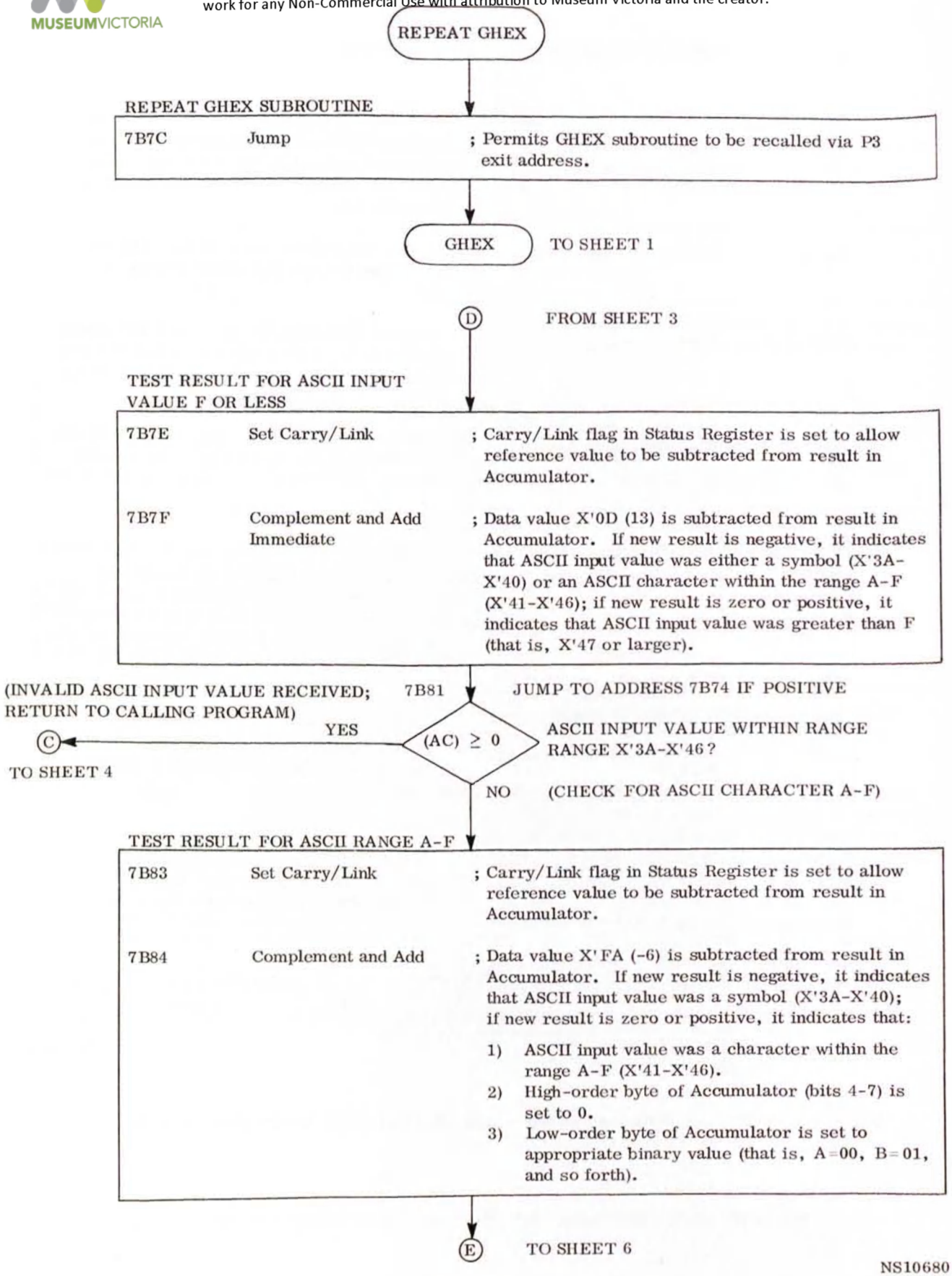


Figure 4-8. GHEX and GHEXE Subroutine — Annotated Instruction Listing (Sheet 5 of 12)

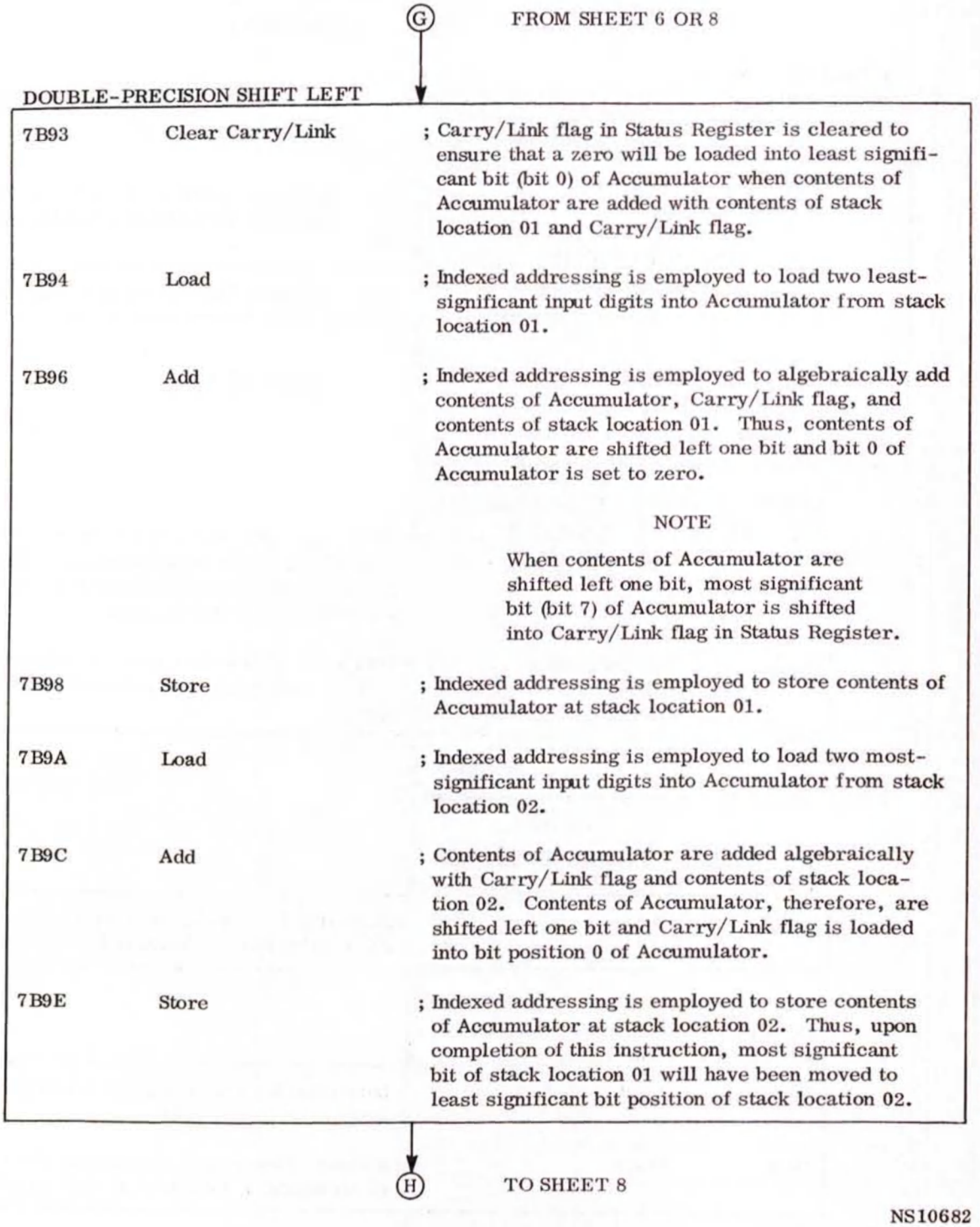


Figure 4-8. GHEX and GHEXE Subroutine — Annotated Instruction Listing (Sheet 7 of 12)

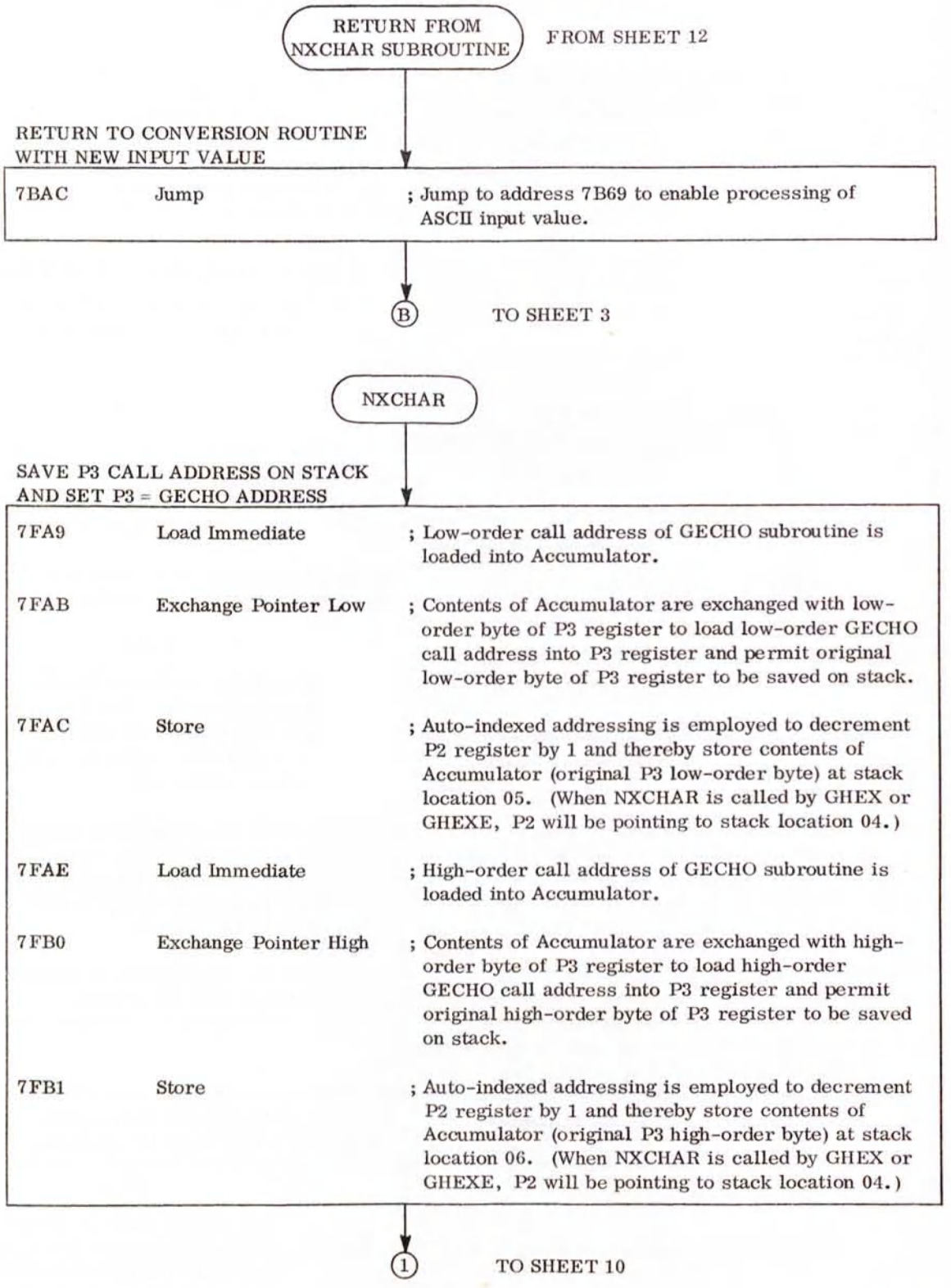


Figure 4-8. GHEX and GHEXE Subroutine — Annotated Instruction Listing (Sheet 9 of 12)

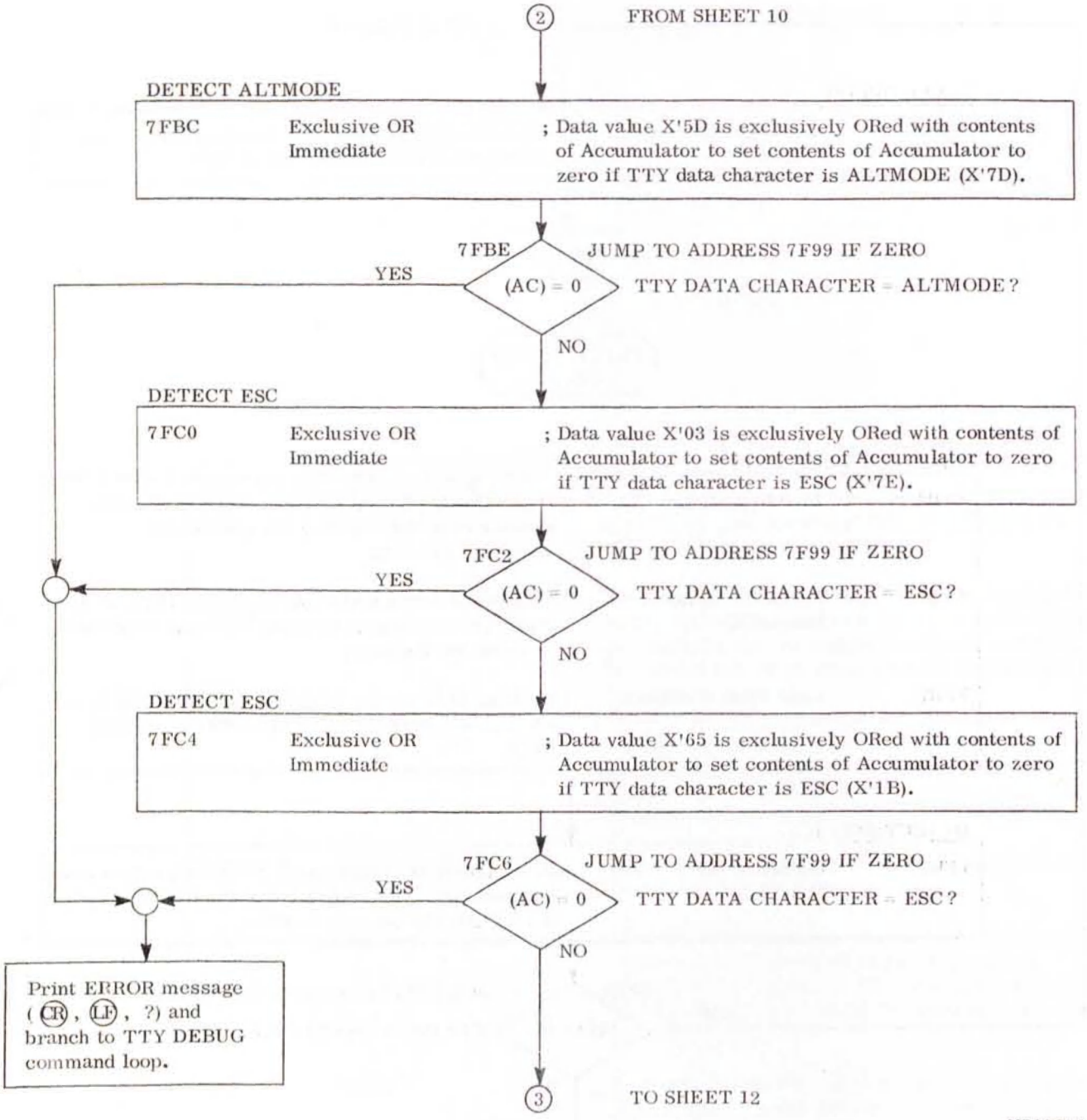


Figure 4-8. GHEX and GHEXE Subroutine — Annotated Instruction Listing (Sheet 11 of 12)

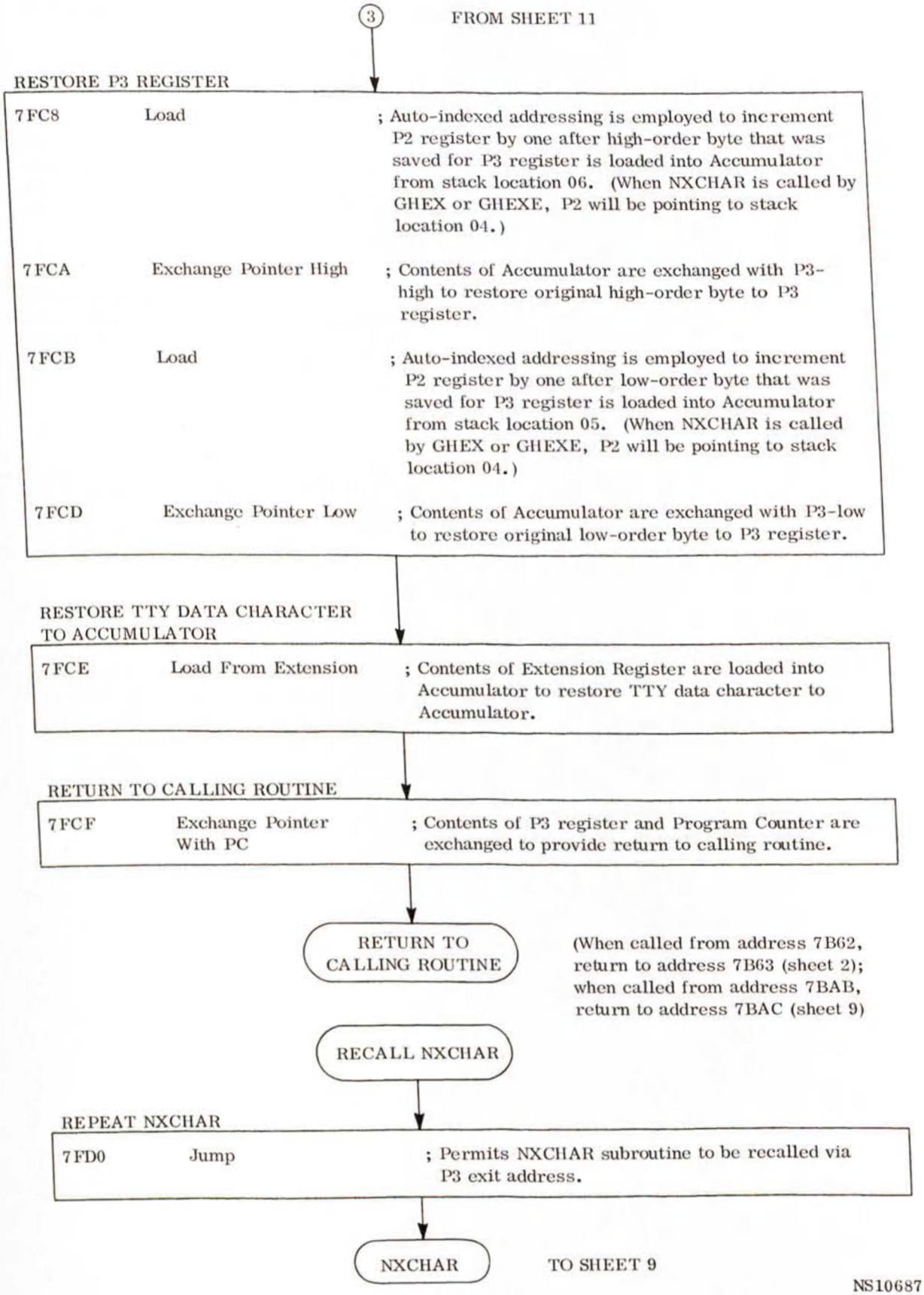


Figure 4-8. GHEX and GHEXE Subroutine — Annotated Instruction Listing (Sheet 12 of 12)

GHEX and GHEXE Subroutine — Actual Listing

```
                        .PAGE   'HEX NUMBER INPUT'
                        .LOCAL
                ;
                ; GHEX GETS A 16-BIT VALUE AND PUSHES IT TO THE STACK.
                ; GHEXE ASSUMES THE FIRST CHAR IS IN THE E REGISTER.
                ; ONLY THE LAST 4 INPUT DIGITS ARE SAVED.
                ;
                ; RETURNS VALUE IN TOP 2 WORDS OF STACK AND TERMINATOR
                ; IN THE AC AND EX REGISTERS.
                ;
7B4C C401       GHEXE:  LDI     1
7B4E 9002               JMP     $6
7B50 C400       GHEX:   LDI     0               ; RESET GHEXE FLAG
7B52 CAFB       $6:     ST      -5(P2)
7B54 C4A8               LDI     L(NXCHAR)-1     ; SAVE RETURN ADDRESS AND SET UP
7B56 33                 XPAL    P3              ;   TO NXCHAR
7B57 CEFD               ST      @-3(P2)         ; STORE RETURN ADDRESS TO LEAVE ROOM
7B59 C47F               LDI     H(NXCHAR)       ;   FOR RESULT
7B5B 37                 XPAH    P3
7B5C CEFF               ST      @-1(P2)
7B5E C2FF               LD      -1(P2)
7B60 9C01               JNZ     $1
7B62 3F                 XPPC    P3
7B63 C400       $1:     LDI     0               ; INITIALIZE RESULT TO 0
7B65 CA03               ST      3(P2)
7B67 CA02               ST      2(P2)
7B69 40         $LOOP:  LDE
7B6A 03                 SCL
7B6B FC3A               CAI     '9'+1           ; CHECK FOR 0-9
7B6D 940F               JP      $2              ; NOT 0-9, TOO LARGE
7B6F 03                 SCL
7B70 FCF6               CAI     '0'-'9'-1       ; CHECK FOR 0-9
7B72 9419               JP      $3              ; IF POSITIVE, NUMBER IS
                                                ;   IN RANGE AND CONVERTED.
7B74 C601       $RET:   LD      @1(P2)          ; NUMBER IS NOT A HEX DIGIT,
7B76 37                 XPAH    P3              ;   RETURN
7B77 C601               LD      @1(P2)
7B79 33                 XPAL    P3
7B7A 40                 LDE
7B7B 3F                 XPPC    P3
7B7C 90D2               JMP     GHEX
7B7E 03         $2:     SCL
7B7F FC0D               CAI     'F'+1-'9'-1     ; CHECK FOR DIGITS A-F.
7B81 94F1               JP      $RET            ; NUMBER TOO LARGE
7B83 03                 SCL
7B84 FCFA               CAI     'A'-'F'-1
7B86 9402               JP      $4              ; DIGIT BETWEEN A&F
7B88 90EA               JMP     $RET
7B8A 02         $4:     CCL
7B8B F40A               ADI     10              ; ADJUST DIGIT VALUE FOR 10-16
7B8D CAFF       $3:     ST      -1(P2)          ; SAVE ADJUSTED DIGIT
7B8F C404               LDI     4               ; SET UP BIT COUNTER FOR
7B91 CAFE               ST      -2(P2)          ;   SHIFT.
7B93 02         $5:     CCL                     ; SHIFT HEX DIGIT LEFT ONE
7B94 C203               LD      3(P2)           ;   DIGIT, ONE BIT EACH
7B96 F203               ADD     3(P2)           ;   TIME THROUGH LOOP.
7B98 CA03               ST      3(P2)
7B9A C202               LD      2(P2)
7B9C F202               ADD     2(P2)
7B9E CA02               ST      2(P2)
7BA0 BAFE               DLD     -2(P2)
7BA2 9CEF               JNZ     $5
7BA4 02                 CCL
7BA5 C203               LD      3(P2)           ; ADD CURRENT DIGIT INTO
7BA7 F2FF               ADD     -1(P2)          ;   NUMBER
7BA9 CA03               ST      3(P2)
7BAB 3F                 XPPC    P3              ; GET NEXT CHAR
7BAC 90BB               JMP     $LOOP           ; AND LOOP
```

A. Set Carry/Link (C/L) flag in Status Register to enable reference value to be subtracted from ASCII input in Accumulator.

B. Subtract data value X'3A (58) from ASCII input in Accumulator via Complement-And-Add Instruction. If result is negative, proceed to C below to test result for ASCII values 0-9; if result is zero or positive, proceed to D below to test that result indicates ASCII value F or less.

	/ (X'2F)	0 (X'30)	9 (X'39)	: X'(3A)
AC	0010 1111	0011 0000	0011 1001	0011 1010
Data	1100 0101	1100 0101	1100 0101	1100 0101
C/L	1	1	1	1
Result	1111 0101	1111 0110	1111 1111	0000 0000
C/L	0	0	0	1

	@ (X'40)	A (X'41)	F (X'46)	G (X'47)
AC	0100 0000	0100 0001	0100 0110	0100 0111
Data	1100 0101	1100 0101	1100 0101	1100 0101
C/L	1	1	1	1
Result	0000 0110	0000 0111	0000 1100	0000 1101
C/L	1	1	1	1

C. Set Carry/Link flag in Status Register, then, subtract data value X'F6 (-10) from contents of Accumulator via Complement-And-Add Instruction. If result is negative (indicating that ASCII input was X'2F or less), detect invalid ASCII input and terminate ASCII-to-hexadecimal conversion; if result is zero or positive, ASCII character was within range 0-9 and conversion is completed.

	/ (X'2F)	0 (X'30)	9 (X'39)
AC	1111 0101	1111 0110	1111 1111
Data	0000 1001	0000 1001	0000 1001
C/L	1	1	1
Result	1111 1111	0000 0000	0000 1001
C/L	0	1	1

D. Set Carry/Link flag in Status Register, then subtract data value X'0D (13) from contents of Accumulator via Complement-And-Add Instruction. If result is zero or positive (indicating that ASCII value was X'47 or greater), detect invalid ASCII input and terminate ASCII-to-hexadecimal conversion; if result is negative, proceed to E (sheet 2) to test that result indicates ASCII value A or greater.

	: (X'3A)	@ (X'40)	A (X'41)
AC	0000 0000	0000 0110	0000 0111
Data	1111 0010	1111 0010	1111 0010
C/L	1	1	1
Result	1111 0011	1111 1001	1111 1010
C/L	0	0	0

	F (X'46)	G (X'47)
AC	0000 1100	0000 1101
Data	1111 0010	1111 0010
C/L	1	1
Result	1111 1111	0000 0000
C/L	0	1

NS10689

Figure 4-10. ASCII-to-Hexadecimal Conversion Routine for GHEX and GHEXE Subroutines (Sheet 1 of 2)

The GHEX and GHEXE subroutines store each valid input at the least-significant-digit stack location after shifting the previous input one digit to the left. To illustrate execution of the 4-digit save routine, the values DFB7 are arbitrarily assumed to have been previously received and stored on the stack as shown below, and the current input value is arbitrarily assumed to be F.

		Bit 7	Bit 0
Previous Input — stack location	01	B	7
	02	D	F

A. Temporarily store current input (X'0F) at stack location 05.

B. Initialize Bit Count to 04 and store bit count at stack location 06.

C. Clear Carry/Link flag in Status Register, load contents of stack location 01 into Accumulator, and then add contents of stack location 01 with contents of Accumulator and Carry/Link flag. In effect, this operation multiplies the contents of stack location 01 by 2, thereby shifting the result in the Accumulator left one bit and retaining the most significant bit in the Carry/Link flag.

	B	7
AC	1011	0111
Stack location 01	1011	0111
C/L		0
Result	0110	1110
C/L	1	

D. Store result in Stack location 01 — Stack location 01 = 0110 1110, C/L = 1

E. Load contents of stack location 02 into Accumulator, then add contents of stack location 02 with contents of Accumulator and Carry/Link flag. In effect, this operation multiplies the contents of stack location 02 by 2, thereby shifting the result in the Accumulator left one bit and loading bit 7 of stack location 01 into bit 0 of stack location 02 via the Carry/Link flag. (Since only the last four digits are saved, the stack location 02 bit 7 value that is loaded into the Carry/Link flag is not retained.)

	D	F
AC	1101	1111
Stack location 02	1101	1111
C/L		1
Result	1011	1111
C/L	1	

F. Store result in stack location 02 — Stack location 02 = 1011 1111

G. DECREMENT bit count in stack location 06.

NS10691

Figure 4-11. Four-Digit Save Routine for GHEX and GHEXE Subroutines (Sheet 1 of 2)

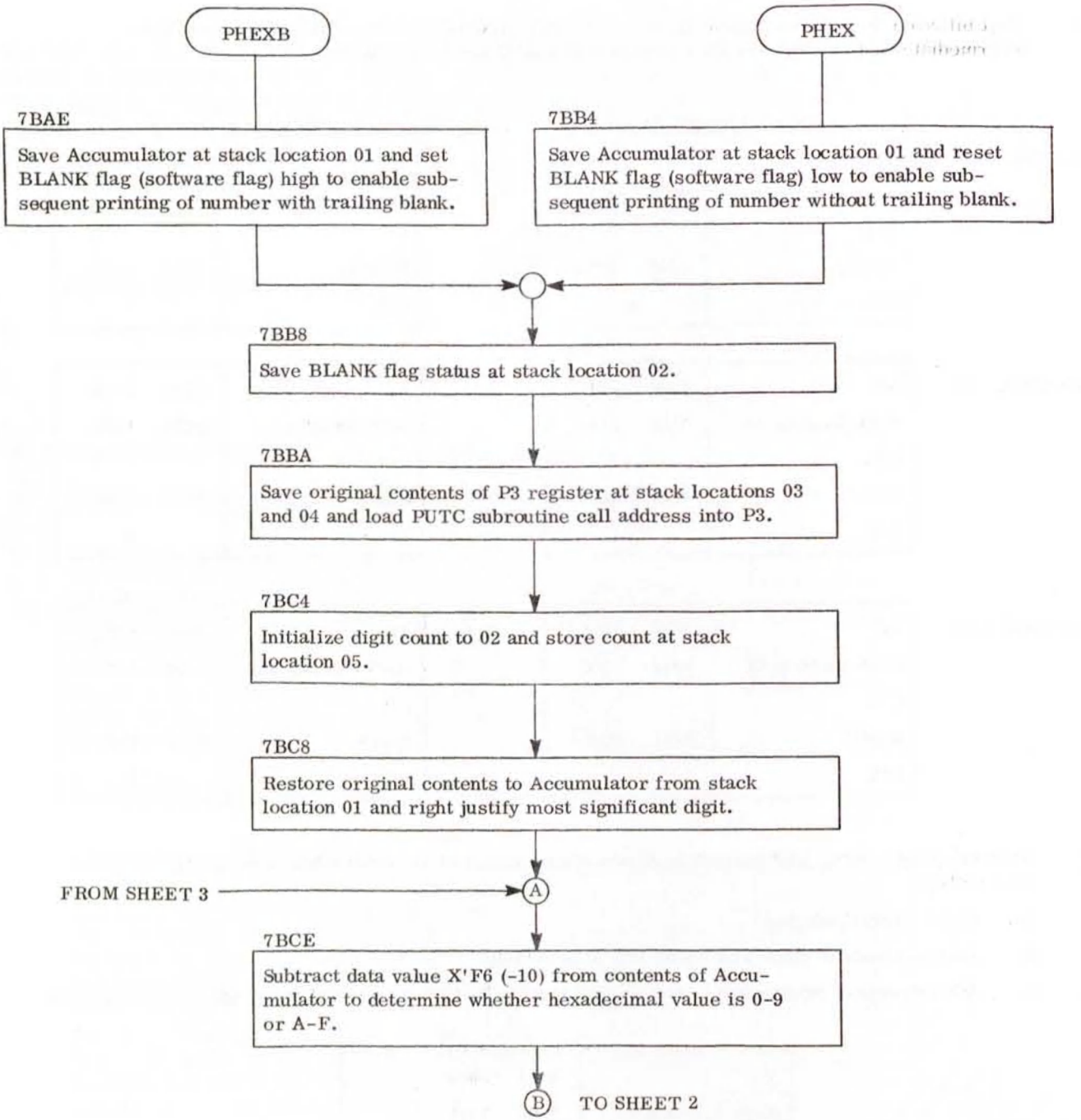
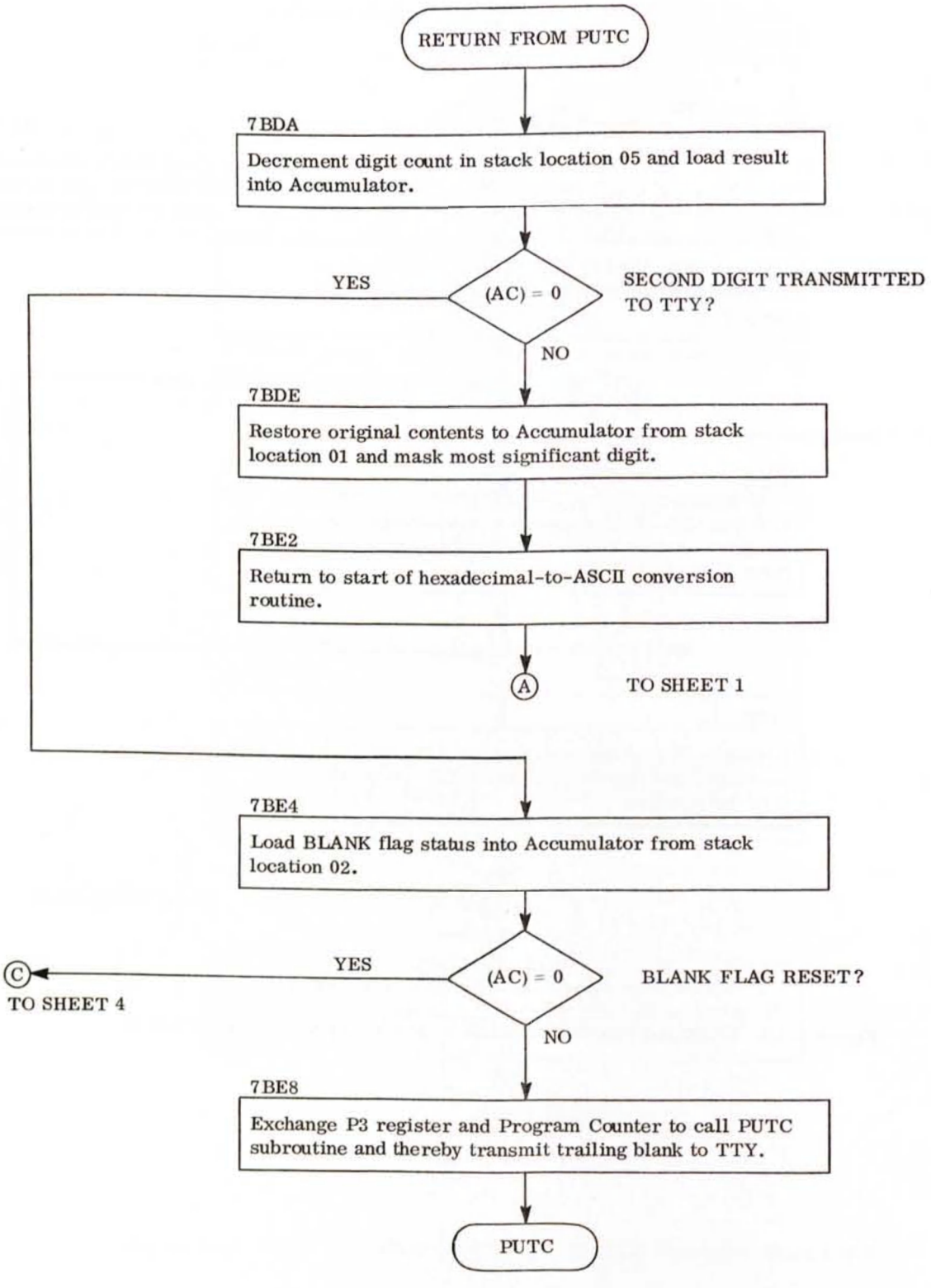


Figure 4-12. PHEX and PHEXB Subroutine — Detailed Flowchart (Sheet 1 of 4)



NS10695

Figure 4-12. PHEX and PHEXB Subroutine — Detailed Flowchart (Sheet 3 of 4)

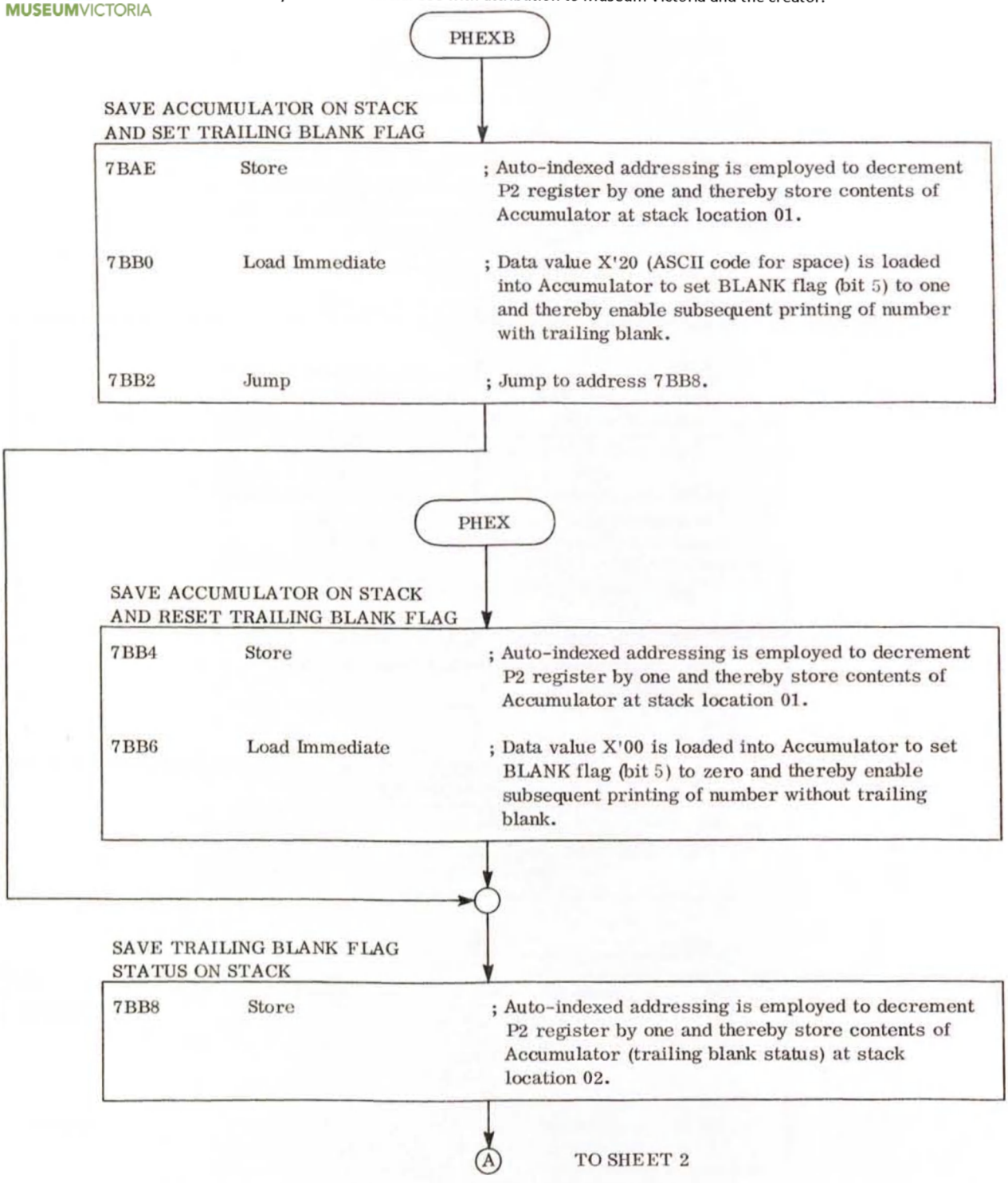


Figure 4-13. PHEX and PHEXB Subroutine — Annotated Instruction Listing (Sheet 1 of 7)

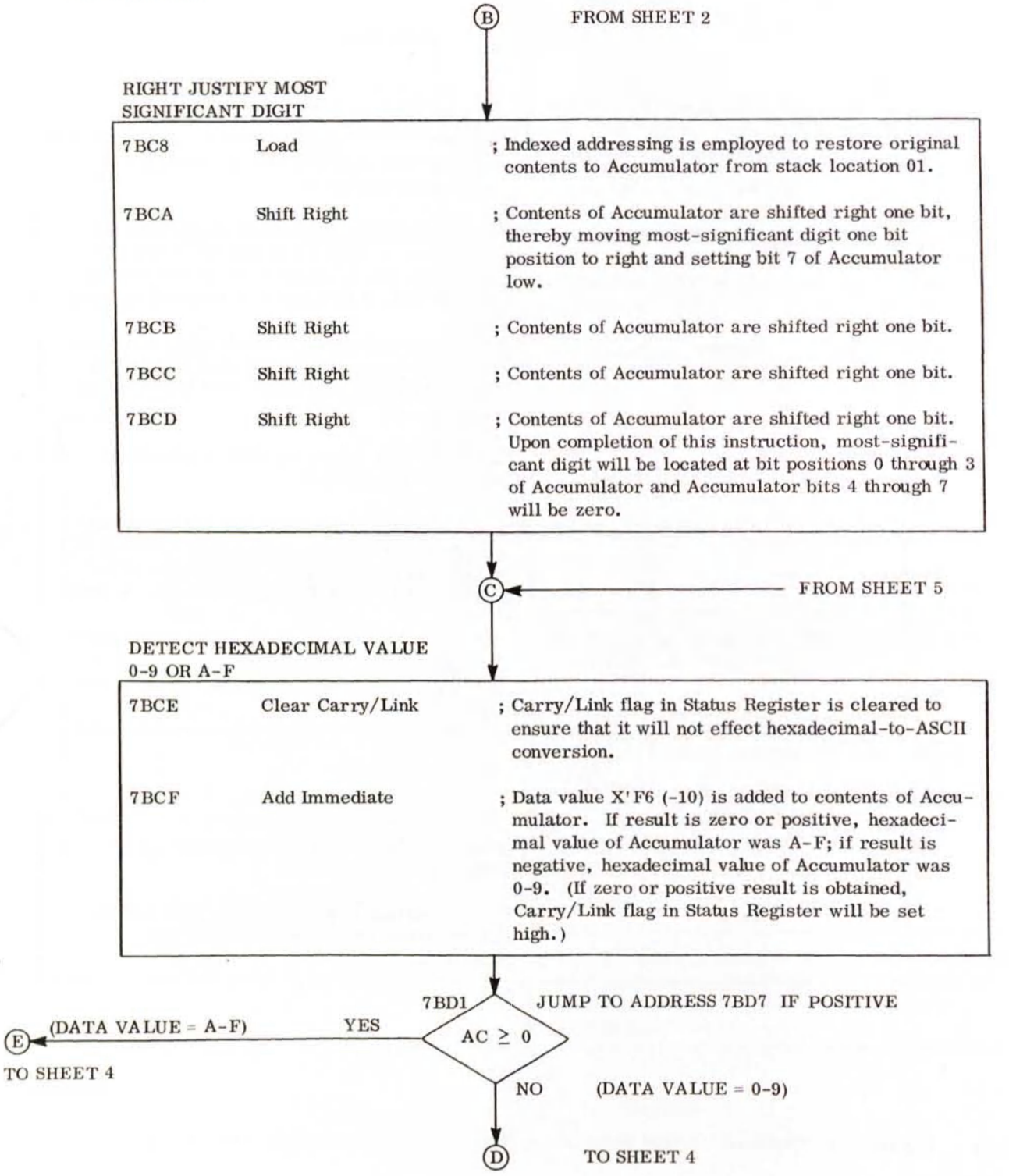


Figure 4-13. PHEX and PHEXB Subroutine — Annotated Instruction Listing (Sheet 3 of 7)

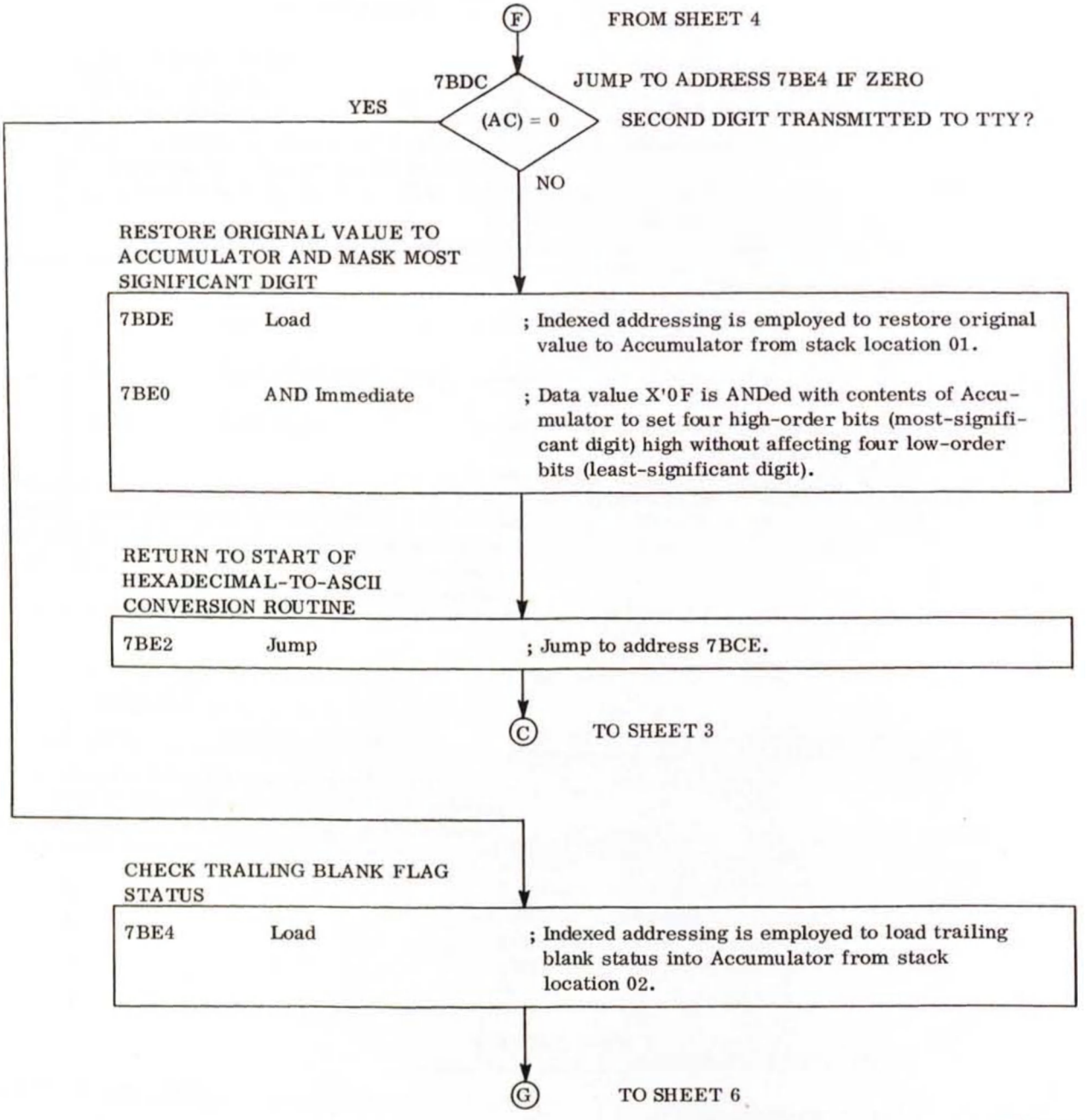
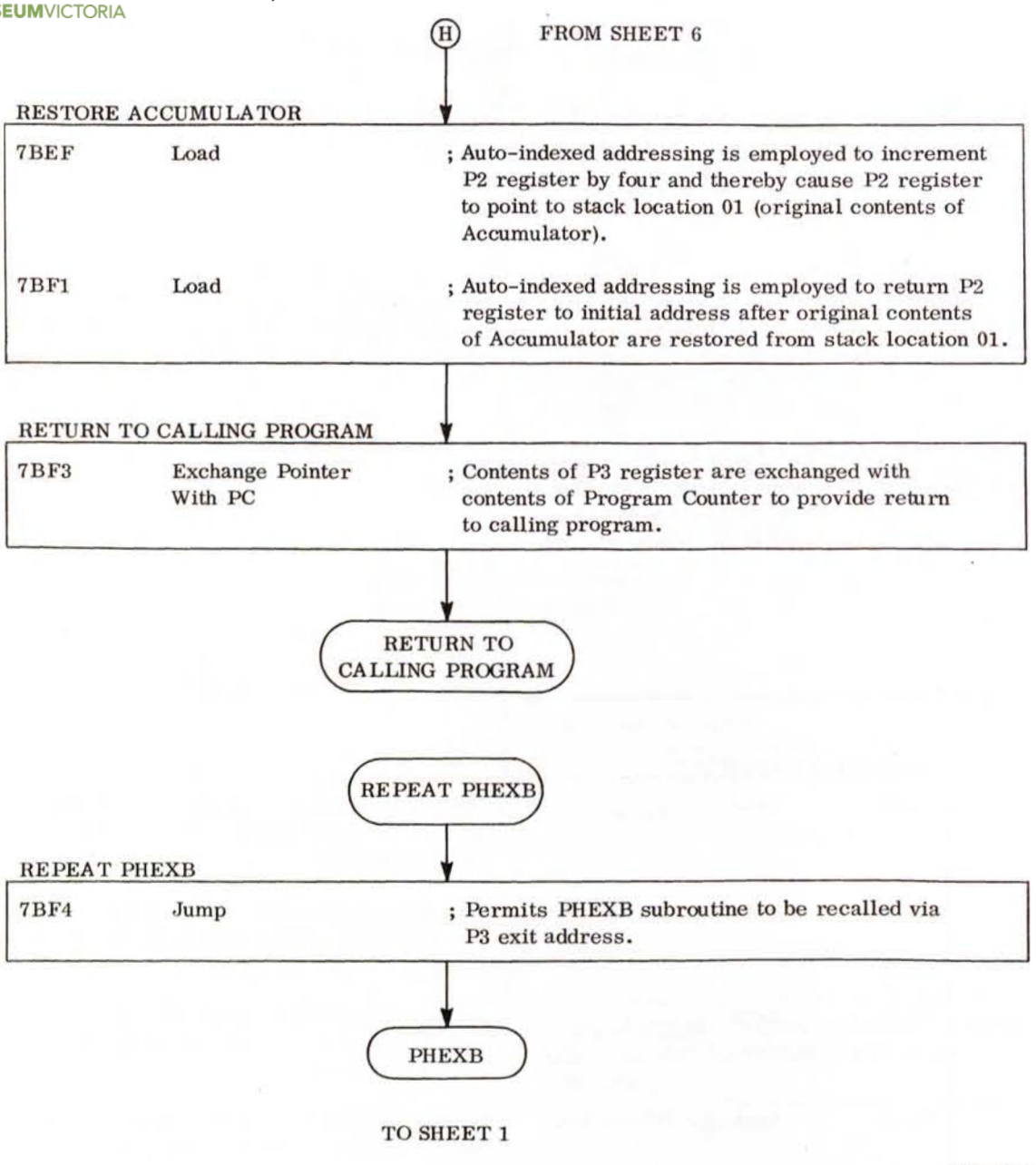


Figure 4-13. PHEX and PHEXB Subroutine — Annotated Instruction Listing (Sheet 5 of 7)



NS10703

Figure 4-13. PHEX and PHEXB Subroutine — Annotated Instruction Listing (Sheet 7 of 7)

Chapter 5

FUNCTIONAL DESCRIPTION

5.1 GENERAL DESCRIPTION

The LCDS (see figure 5-1) provides control and display functions that permit an SC/MP CPU Application Card to be used to develop, test, and debug SC/MP hardware and software applications designs. Functionally, the LCDS consists of a resident firmware program, a keyboard and display panel (Panel), and all of the logic and buffer circuits required to transfer control back and forth between the resident firmware program (DEBUG Mode) and a user-entered applications program (RUN Mode). From a software standpoint, DEBUG operations encompass the address range 7000 through 7FFF; the remainder of the 64K address range (0000 through 6FFF and 8000 through FFFF) is available for storage of the user's applications program and assignment of RUN Mode functions to the user's applications hardware.

When the DEBUG Mode is selected, it is entered via a hardware-forced Save Routine that is executed when the current RUN Mode instruction is completed. During the Save Routine, the contents of the SC/MP accumulator, status and P3 registers are stored in LCDS memory and the P3 register is loaded with address 7808. Then, as the last step of the Save Routine, the contents of the P3 register and program counter are exchanged so that execution of the resident DEBUG firmware program can be initiated at starting address 7809.

At the start of the DEBUG firmware program, the contents of the remaining SC/MP registers are saved in LCDS memory, then the setting of the HALT MODE switch is checked to determine whether control and display capability is to be provided at the Panel or at the optional TTY. When the HALT MODE switch is set to PNL, the DEBUG firmware branches to the Panel Service Routines and displays the saved PC value on the digital readout to indicate that DEBUG commands may be entered via the panel keyboard; similarly, when the HALT MODE switch is set to TTY, the DEBUG firmware branches to the TTY Service Routines and prints out the saved PC contents on the TTY to indicate that DEBUG commands may be entered via the TTY keyboard. After displaying or printing out the saved PC value, the DEBUG firmware then continually cycles through the appropriate service routines to permit processing of ensuing DEBUG commands as described in chapter 3, Operation.

Termination of the DEBUG Mode occurs when the RUN Mode is subsequently selected via the RUN pushbutton (HALT MODE switch set to PNL) or the TTY GO command (HALT MODE switch set to TTY). After the RUN Mode is selected, the DEBUG firmware branches to an Exit Subroutine that restores part of the saved status to the SC/MP microprocessor; then, a hardware-forced DEBUG Return Routine is executed to restore the remaining status to the SC/MP microprocessor. Upon completion of the DEBUG Return Routine, an IFETCH data input/output cycle occurs to initiate RUN Mode operation at the restored PC address and thereby provide a return to the user's application program without loss of SC/MP microprocessor status. (Users familiar with operation of the SC/MP CPU Application Card should note that the DEBUG Return Routine decrements the value contained in memory locations 77F5 and 77F6 before restoring it to the PC. This action compensates for the automatic PC increment which occurs during the IFETCH data input/output cycle that initiates RUN Mode operation.)

NOTE

The following paragraphs assume a thorough understanding of the operation of the SC/MP CPU Application Card. For detailed information on the operation of the SC/MP CPU Application Card, refer to the SC/MP CPU Application Card Specification.

5.1.1 CPU and Data I/O Control

As shown in figure 5-1, the LCDS provides a prewired interface bus at connectors J1 through J4 to permit plug-in interconnection of an SC/MP CPU Application Card and other SC/MP family application cards. Wiring of the bus is such that the basic CPU control signals (MEMRDY, RUN, and ENCPU) are pulled to +5V via resistors on the LCDS to permit continuous operation of the CPU card when the DEBUG Mode is enabled, and application-system control of CPU card operation when the RUN Mode is enabled. Similarly, the MEMEN (Address Enable) input for the SC/MP family application cards is also pulled to +5V so that the address compare circuit on the cards will remain continuously enabled except when the MEMEN signal is externally driven low for special-purpose applications.

TRI-STATE® buffers are employed for interconnection of the Buffered Address and Data Buses to permit program-controlled operation of the LCDS via the SC/MP CPU Application Card. Thus, the DEBUG Address Select* output of the Mode Control Logic is set high, except during the DEBUG Save and Restore Routines, so that the Address Buffers will continuously drive the address output of the SC/MP CPU Application Card onto the LCDS Address Bus. (Mode Control Logic and Address Buffer operation for the DEBUG Save and Restore Routines is covered in the paragraphs that follow.) Whenever the address output of the SC/MP CPU Application Card is within the range assigned to the LCDS (7000 through 7FFF), the Address Decode Logic drives the MEMSEL* output low for address acknowledgement and provides an enable signal to the appropriate memory device (addresses 7600 through 7FFF) or the Keyboard and Display Panel (address 7000 through 71FF). The ensuing BRDS* or BWDS* strobe output of the SC/MP CPU Application Card then causes data to be written into or read out of the addressed location.

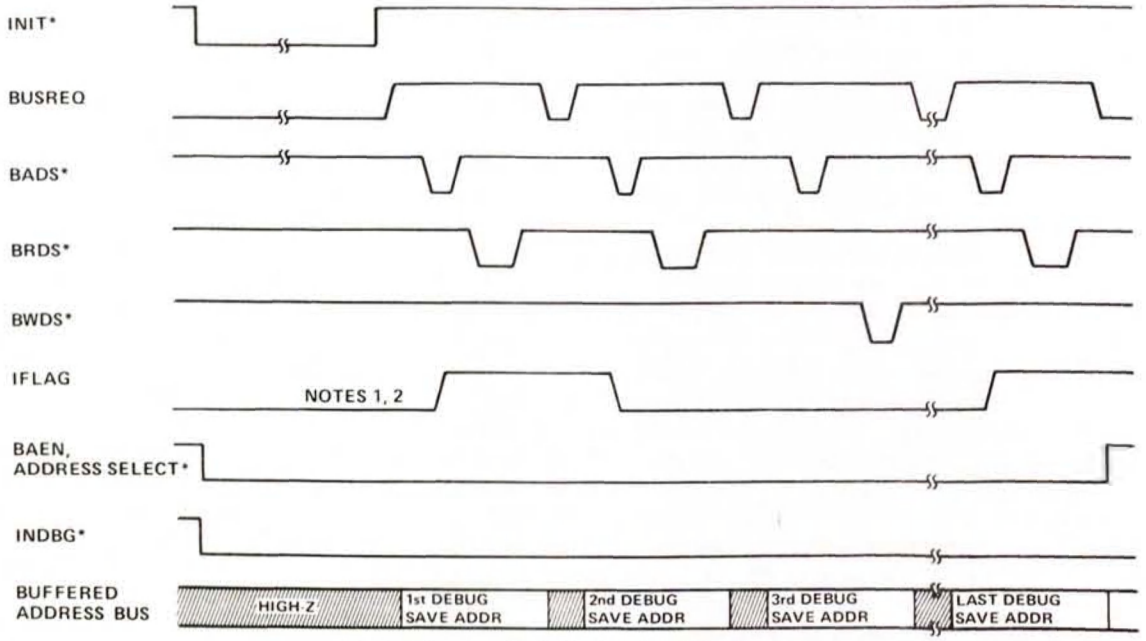
Gating of read/write data to/from the Buffered Data Bus occurs under control of the BRDS* strobe. When the BRDS* strobe goes low while the LCDS is addressed, the Data Buffers are configured to drive data from the DEBUG Data Out Bus onto the Buffered Data Bus from which the SC/MP CPU Application Card accepts the data on the trailing edge of the BRDS* strobe. At all times, the Data Buffers are configured to drive data from the Buffered Data Bus onto the DEBUG Data In Bus so that the data can be written into an addressed LCDS location by the BWDS* strobe.

5.1.2 Hardware-Forced DEBUG Save Routine

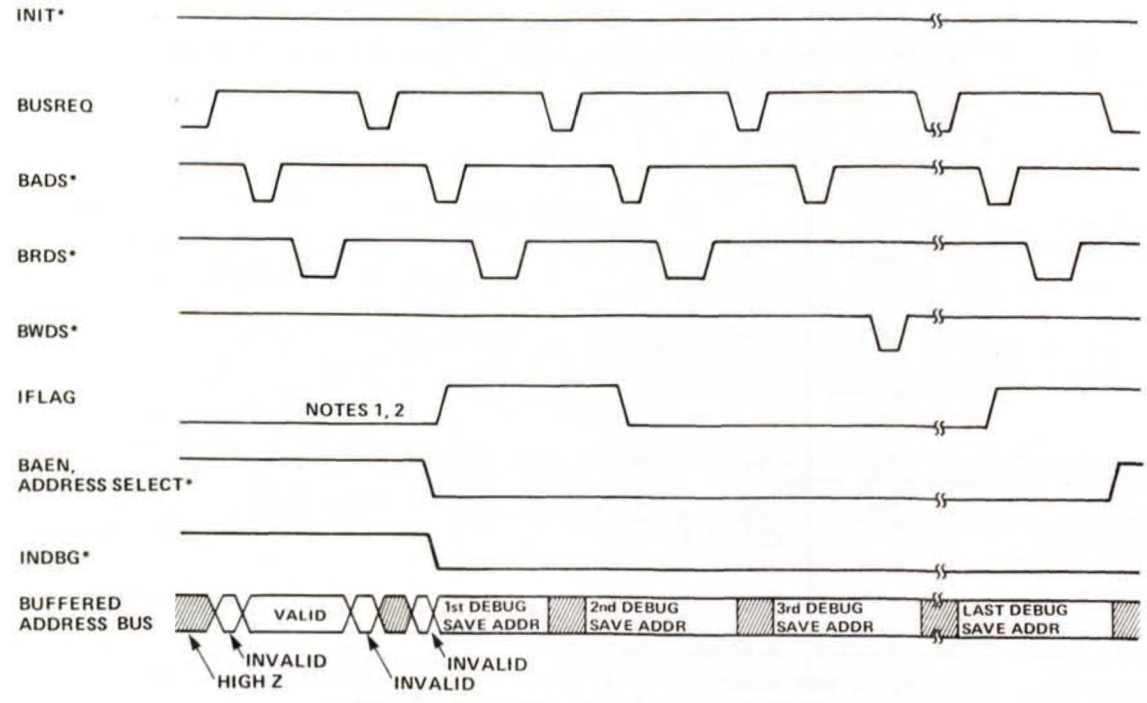
The DEBUG Save Routine is a hardware-controlled subroutine that is executed following initialization of the LCDS or selection of the DEBUG Mode. The Save Routine functions in conjunction with the first subroutine of the DEBUG firmware to cause the internal status of the SC/MP microprocessor to be saved in LCDS memory locations 77F5 through 77FF before the firmware branches to the Panel or TTY Service Routines. During execution of the Panel or TTY Service Routines, the values saved for the SC/MP program counter, registers, and accumulator can then be examined and/or modified as desired.

When power is first applied to the LCDS, the DEBUG Save Routine is executed in response to the low-going INIT* output of the SC/MP CPU Application Card, thereby causing the LCDS to power-up in the DEBUG Mode. Following the power-up initialization sequence, execution of the DEBUG Save Routine reoccurs for any of the following specific conditions:

1. The INIT switch is pressed.
2. The HALT switch is pressed while RUN Mode operation is enabled. (Execution of the DEBUG Save Routine is initiated after the instruction in progress is completed).
3. The RUN/STEP switch is set to STEP and one instruction is executed after the RUN Mode is selected.
4. The HALT INST switch is set to DEBUG and a Halt Instruction is executed while RUN Mode operation is enabled.
5. The DEBUG input to the LCDS is externally driven low while RUN Mode operation is enabled.



A. Debug Save Routine Initiated by Low-Going INIT* Pulse



B. Debug Save Routine Not Initiated by Low-Going INIT* Pulse

NOTES:
1. The IFLAG output of the SC/MP CPU Application Card is shown only as a general timing reference; it is not used by the LCDS.
2. The Mode Control Logic detects an IFETCH data I/O cycle by processing the DI5 data-bit input under control of the BADS* strobe.

NS10706

Figure 5-2. DEBUG Save Routine Timing

After the last DEBUG Save Routine Address is output, the BAEN and DEBUG Address Enable* outputs of the Mode Control Logic are reset high on the trailing edge of the BUSREQ signal to reenable the address outputs of the SC/MP CPU Application Card. On the leading edge of the next BUSREQ signal, the SC/MP CPU Application Card then outputs address 7809 to initiate execution of the DEBUG firmware program. The first routine of the DEBUG firmware program, in turn, saves the contents of the remaining SC/MP registers (PC, P1, P2, and EX) in the designated LCDS memory locations, reads the setting of the TTY HALT MODE switch, and effects a branch to the Panel or TTY Service Routines as appropriate. The Panel or TTY Service Routines are then executed continuously to provide control and display capabilities at the Panel or TTY — as described in chapter 3, Operation.

Users familiar with SC/MP microprocessor operation should note that saving of the program counter contents is effected mathematically rather than directly. When the DEBUG Save Routine is initiated, the SC/MP program counter will have already output the address of the next instruction that would have been fetched had the DEBUG Mode not been selected. Incrementing of the program counter then continues normally during the DEBUG Save Routine and, when the XPPC P3 instruction is executed to complete the DEBUG Save Routine, the program counter will have been incremented 15 times. (As shown in figure 4 of the SC/MP CPU Application Card Data Sheet, incrementing of the program counter occurs at the start of each data input/output cycle except when an Effective Address is output during a data input/output cycle.) During the first DEBUG firmware routine, therefore, the value 15 is subtracted from the P3 register, and the result is stored in LCDS memory locations 77F5 and 77F6. Thus, the value saved for the program counter is the address that was present on the Buffered Address Bus when the DEBUG Save Routine was initiated.

5.1.3 DEBUG Mode Termination

In addition to providing control and display functions, the DEBUG firmware Panel and TTY Service Routines also check for selection of the RUN Mode via the RUN switch or the TTY GO command. When the RUN Mode is selected, it causes the DEBUG firmware to branch to an Exit Routine that initiates restoration of the saved values to the SC/MP microprocessor. During the Exit Routine, the following specific actions occur:

1. The values saved for the extension, P1 and P2 registers are restored to the SC/MP microprocessor.
2. The value saved for the program counter is loaded into the P3 register for temporary storage, then it is decremented by ten to compensate for the automatic incrementing that will occur during the hardware-forced DEBUG Return Routine.
3. The value saved for the status register is loaded into the accumulator for temporary storage and a Halt Instruction is executed to transfer program control to the hardware-forced DEBUG Return Routine.

5.1.4 Hardware-Forced DEBUG Return Routine

The DEBUG Return Routine is a hardware-controlled subroutine that is executed by the Mode Control Logic in response to the Halt Instruction that occurs at the end of the DEBUG firmware Exit Routine. As shown in figure 5-3, execution of the DEBUG Return Routine is similar to execution of the DEBUG Save Routine described previously; that is, the BAEN and DEBUG Address Enable* outputs of the Mode Control Logic are held low while the BUSREQ output of the SC/MP CPU Application Card is processed to force a series of DEBUG Return Routine Addresses onto the Buffered Address Bus (refer to table 5-2). A general summary of the functions provided by the DEBUG Return Routine is as follows:

1. The decremented program counter value that was temporarily stored in pointer register 3 is returned to the program counter so that it will be automatically incremented to the appropriate RUN Mode starting value during further execution of the hardware-forced DEBUG Return Routine.
2. The contents of the accumulator are copied into the status register to restore the saved value to the status register.
3. Pointer Register 3 and the accumulator are loaded from LCDS memory to complete restoration of SC/MP microprocessor internal status.

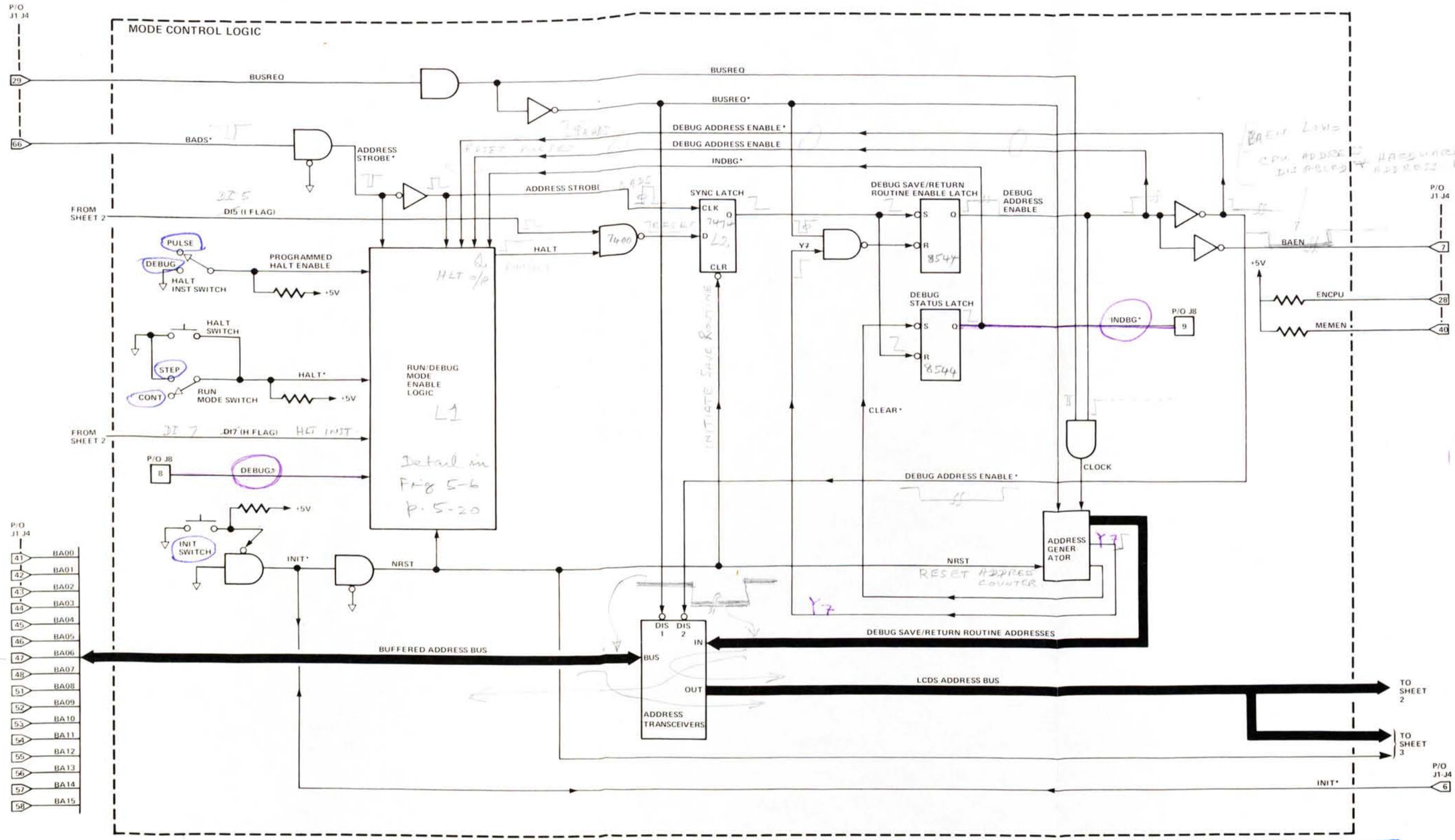
Table 5-2. Hardware-Forced DEBUG Return Routine Address Functions

Instruction	DEBUG Return Address	Data I/O Cycle	Buffered Read/Write Data	Comments
XPPC P3	1. 7808	Read (IFETCH)	3F	SC/MP microprocessor reads in opcode for XPPC P3 Instruction; contents of SC/MP Program Counter are exchanged with contents of Pointer Register 3.
Copy AC to Status (CAS)	2. 7807	Read (IFETCH)	07	SC/MP microprocessor reads in opcode for Copy AC to Status (CAS) Instruction; contents of accumulator are copied into Status Register.
Load Immediate (LI)	3. 7801	Read (IFETCH)	C4	SC/MP microprocessor reads in opcode for Load Immediate (LI) Instruction.
	4. 77FC	Read	Value Saved for P3-low	SC/MP Accumulator is loaded with Pointer Register 3 low-order byte.
XPAL P3	5. 7803	Read (IFETCH)	33	SC/MP microprocessor reads in opcode for XPAL P3 Instruction; contents of SC/MP Accumulator are exchanged with low-order byte of Pointer Register 3.
Load Immediate (LI)	6. 7801	Read (IFETCH)	C4	SC/MP microprocessor reads in opcode for Load Immediate (LI) Instruction.
	7. 77FB	Read	Value Saved for P3 high	SC/MP Accumulator is loaded with Pointer Register 3 high-order byte.
XPAH P3	8. 7805	Read (IFETCH)	37	SC/MP microprocessor reads in opcode for XPAH P3 Instruction; contents of SC/MP Accumulator are exchanged with high-order byte of Pointer Register 3.
Load Immediate (LI)	9. 7801	Read (IFETCH)	C4	SC/MP microprocessor reads in opcode for Load Immediate (LI) Instruction.
	10. 77FD	Read	(AC)	SC/MP Accumulator is loaded with saved value.

5.2 DETAILED DESCRIPTION

A detailed block diagram of the LCDS is provided in figure 5-4 and equivalent-level circuit descriptions follow.

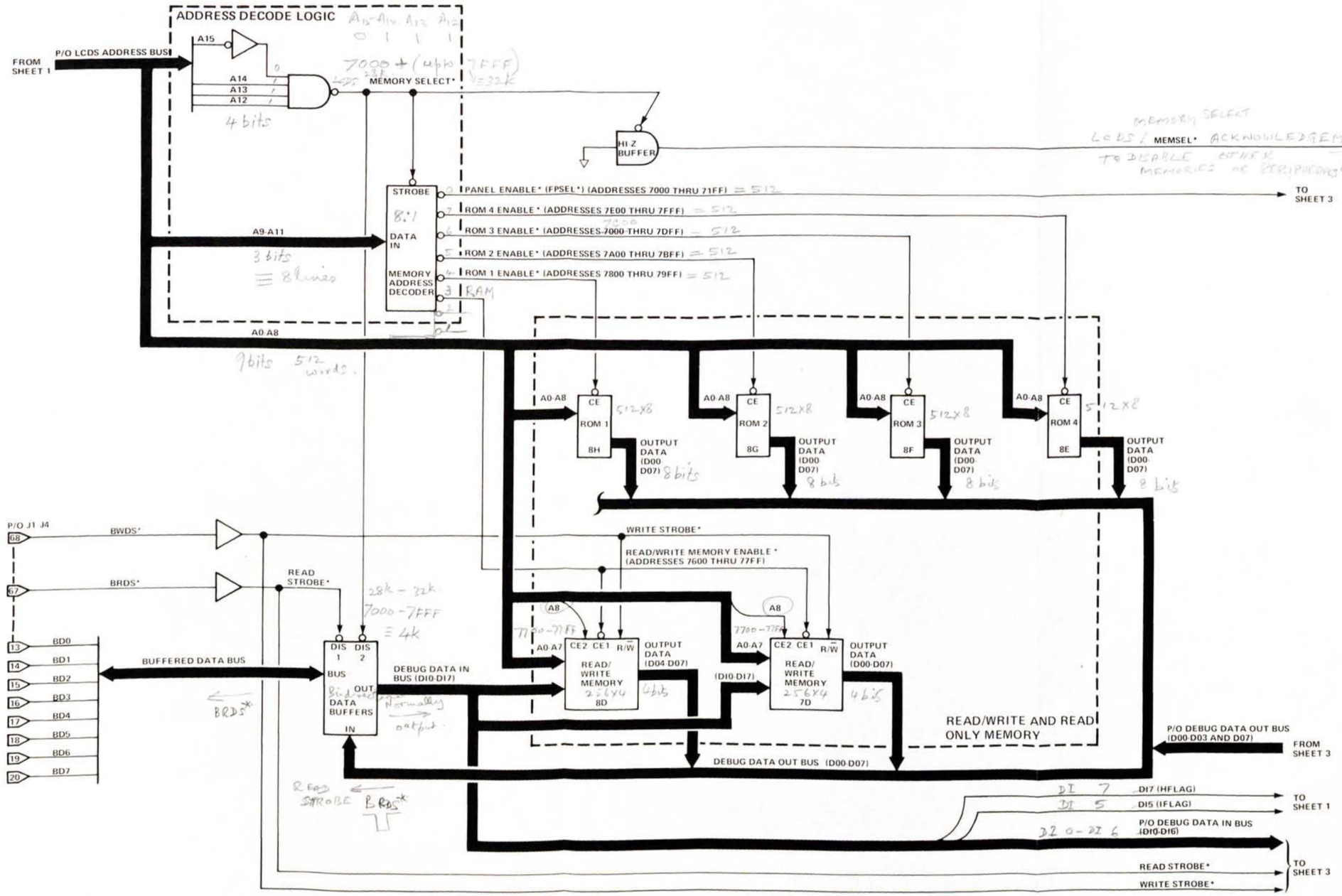
This digitised copy has been provided by Museum Victoria, free of charge. You may use this work for any Non-Commercial Use with attribution to Museum Victoria and the creator.



MODE CONTROL LOGIC NS10708

Figure 5-4. LCDS Block Diagram (Sheet 1 of 3)

This digitised copy has been provided by Museum Victoria, free of charge. You may use this work for any Non-Commercial Use with attribution to Museum Victoria and the creator.

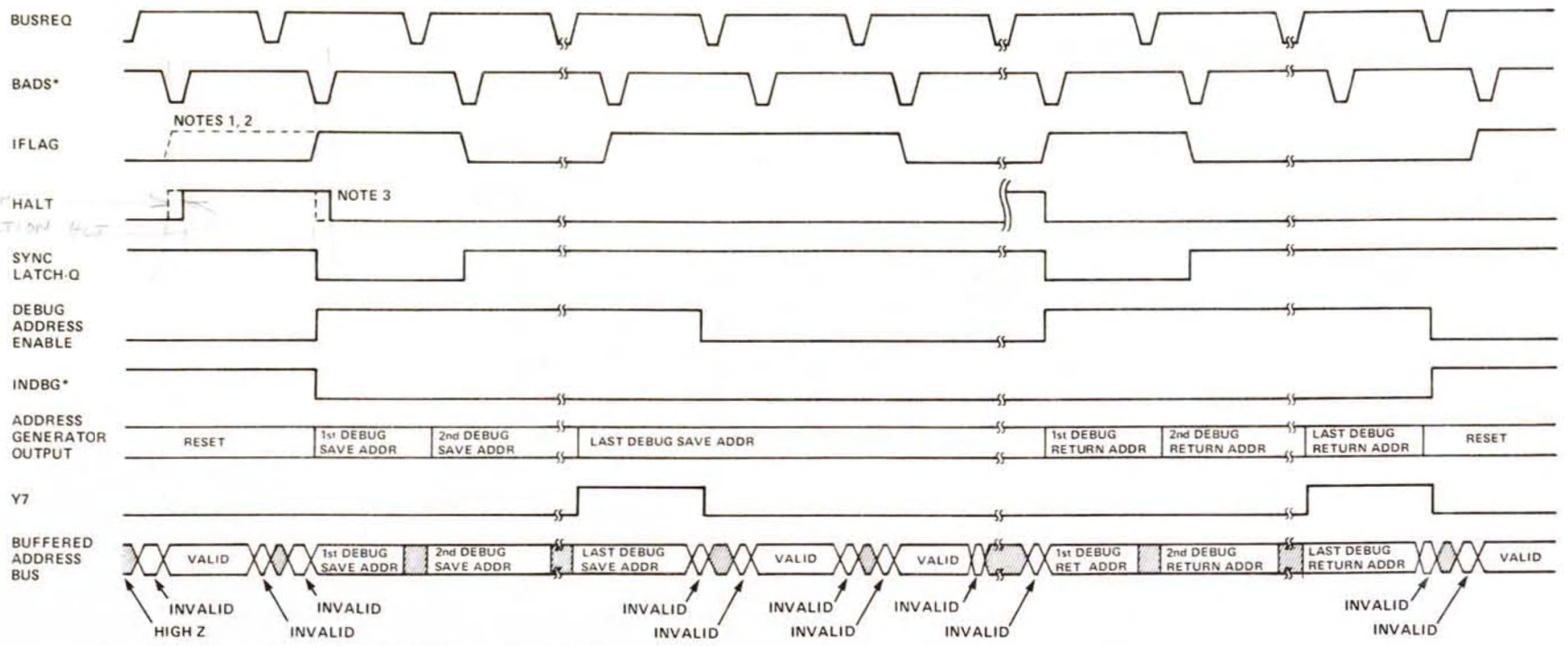


NS10709

Figure 5-4. LCDS Block Diagram (Sheet 2 of 3)

2

FIRMWARE DECODERS



NOTES:
1. The IFLAG output of the SC/MP CPU Application Card is shown only as a general timing reference; it is not used by the LCDS.
2. The IFLAG output of the SC/MP CPU Application Card is set high whenever an IFETCH data I/O cycle is enabled (that is, signal byte instruction or first byte of double-byte instruction is being fetched from memory).
3. When the RUN Mode is selected or the DEBUG Mode is selected via execution of a Halt Instruction, the Halt signal is set high and returned low on the leading edge of the BADS* strobe; when the DEBUG Mode is selected by externally driving the User DEBUG* input low, pressing the Halt switch, or setting the RUN MODE switch to STEP, the Halt signal is set high and returned low on the trailing edge of the BADS* strobe.

NS10711

Figure 5-5. Mode Control Logic Timing for DEBUG Save and Restore Routines

2. The RUN input to the SC/MP CPU Application Card is externally driven low to suspend SC/MP microprocessor operation.

3. The MEMRDY input to the SC/MP CPU Application Card is externally driven low to extend a read or write data input/output cycle.

5.2.7 Keyboard

The keyboard consists of an 18-pushbutton matrix that is enabled under program control via the A0 through A4 address-bit inputs. Thus, when any of these bits goes high, a switch closure in the corresponding row causes a low output to be applied to the Panel Switch Buffers. The Panel Switch Buffers, however, remain disabled until both the Panel Enable* and Read Strobe* inputs go low during the read data input/output cycle of a Load from Address 7000-71FF Instruction. When both the Panel Enable* and Read Strobe* signals are low, the Panel Switch Buffers then invert the data present at the IN terminals and drive the inverted data onto the DEBUG Data Out Bus. For a description of how the Keyboard pushbutton matrix may be employed by a user's application program, refer to 3.6.3.

5.2.8 HALT MODE Switch

In addition to the inputs received from the Keyboard pushbutton matrix, the Panel Switch Buffers also receive a continuous input from the HALT MODE switch. This input is periodically checked by the DEBUG firmware program so that DEBUG command capability will be provided at the selected location (Panel or optional TTY) as described in chapter 3, Operation.

5.3 CIRCUIT DESCRIPTION

Operation of the RUN/DEBUG Mode Enable Logic and the Address Generator circuits is described in the paragraphs that follow.

5.3.1 RUN/DEBUG Mode Enable Logic

The function of the RUN/DEBUG Mode Enable Logic (see figure 5-6) is to indicate a valid RUN or DEBUG Mode selection by providing a high Halt output to the Sync Latch. Thus, when power is first applied to the LCDS, flip-flop 8B is reset by the low NRST signal during the power-up initialization cycle, and the resultant Halt output of gate 6B3 ensures that the LCDS powers up in the DEBUG Mode. Termination of the Halt output then occurs on the trailing edge of the first Address Strobe* (BADS*) that is generated by the SC/MP CPU Application Card after the power-up initialization cycle is completed. (When the first Address Strobe* input is received, the INDBG* status signal already will have been set low, thereby causing a high level to be present at the D input of flip-flop 8B; flip-flop 8B, therefore, will be clocked to the set state on the low-to-high transition of the Address Strobe*.)

After the power-up initialization cycle is completed, latch 9B functions to detect selection of the RUN or DEBUG Mode via execution of a Halt Instruction, and flip-flop 8B functions to detect selection of the DEBUG Mode via the INIT, HALT or RUN MODE switches, or via the external User DEBUG* Input. The specific conditions under which flip-flops 8B and 9B are set and reset are described below.

1. Flip-Flop 8B. After the power-up initialization cycle is completed, flip-flop 8B is clocked to the reset state each time that the DEBUG Mode is selected by pressing the HALT switch, externally driving the User DEBUG* Input low, or setting the RUN MODE switch to STEP. Any of these actions causes a high Halt Enable output to be provided by gate 6B1 and, if the INDBG* status signal is also high, a low output to be provided by gate 6B2. (Gating of the Halt Enable and INDBG* signals together ensures that any of the specified actions will not affect execution of the DEBUG firmware program.) When the output of gate 6B2 goes low, flip-flop 8B is clocked to the reset state on the next low-to-high transition of the Address Strobe*. After flip-flop 8B is clocked to the reset state, the INDBG* status signal is set low in response to the Halt output of gate 6B3 (refer to 5.2.1), then the Halt

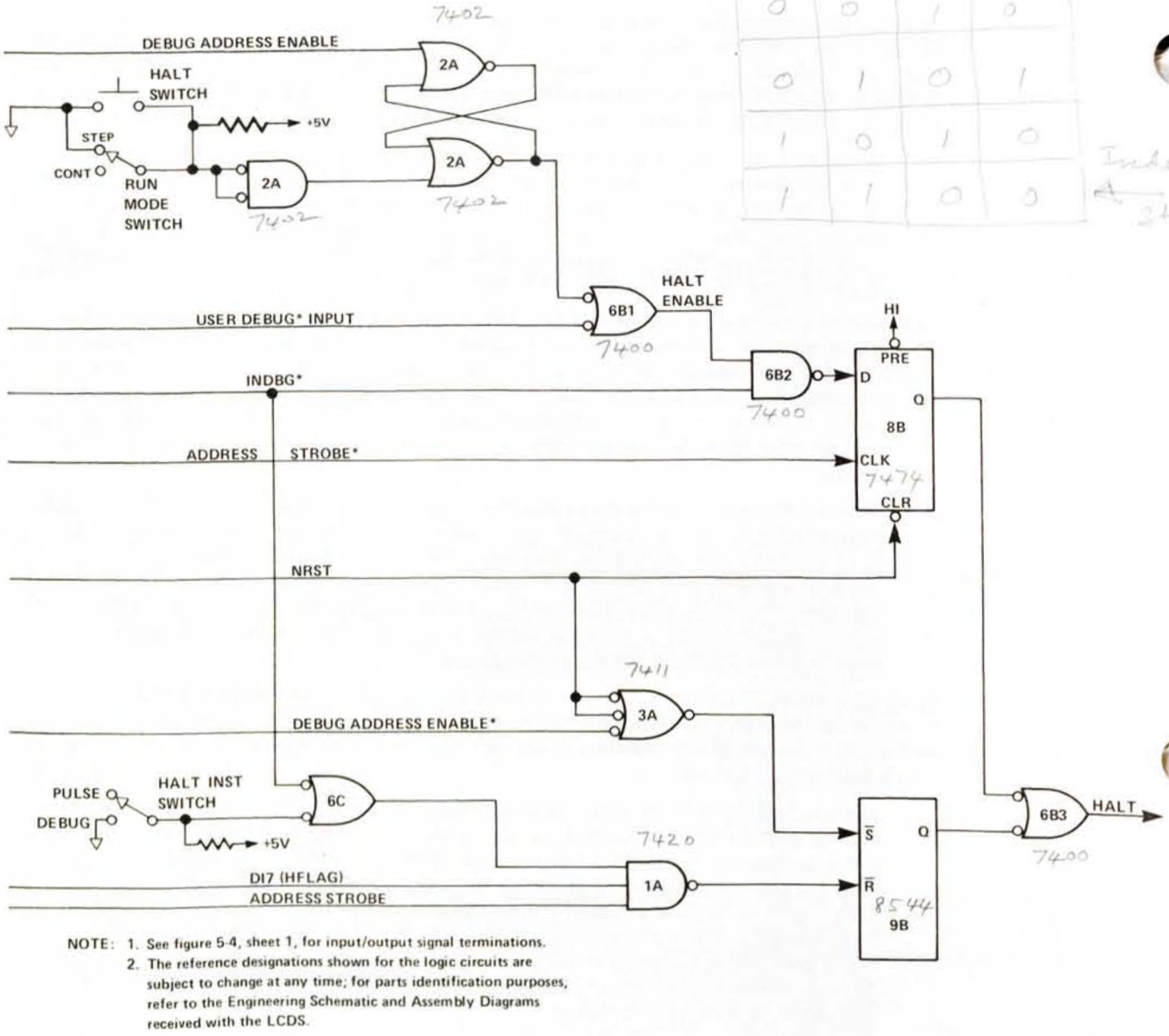
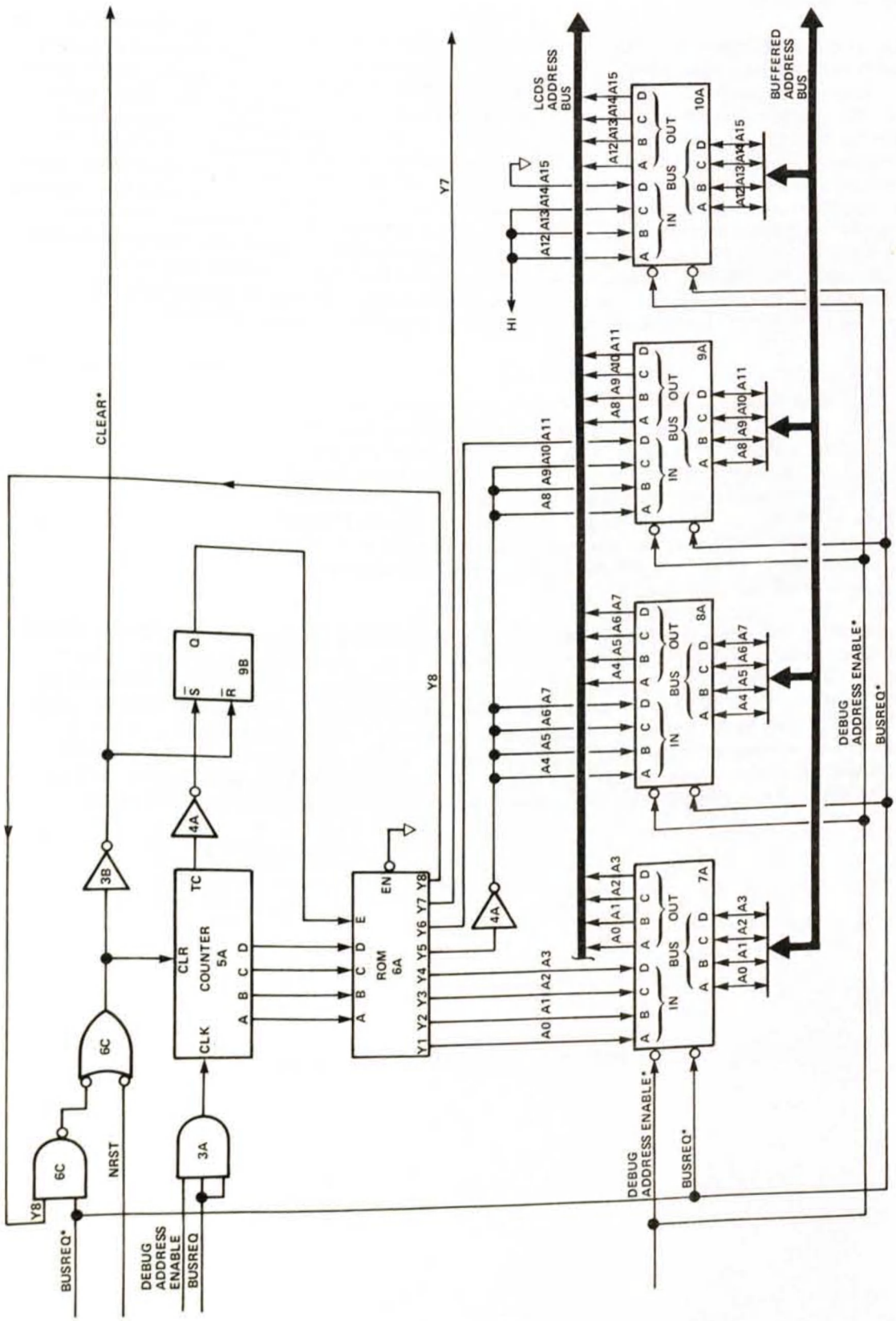


Figure 5-6. RUN/DEBUG Mode Enable Logic Circuit Diagram



NS10713

NOTES:
1. See figure 5-4, sheet 1, for input/output signal terminations.
2. The reference designations shown for the logic circuits are subject to change at any time; for parts identification purposes, refer to the Engineering Schematic and Assembly Diagrams received with the LCDS.

Figure 5-7. DEBUG Address Generator and Address Buffer Circuit Diagram

Chapter 6

SERVICE INFORMATION

Since the LCDS is a sophisticated software-controlled instrument, the types of indications observed for a given malfunction may appear to be totally unrelated to the actual cause of the malfunction. Thus, if an erroneous indication is detected while operating the LCDS, disconnect all applications system circuitry and verify the basic operation of the LCDS according to the Panel and TTY checkout procedures provided in chapter 2, Installation. If all of the indications can be obtained as specified for the Panel and TTY checkout procedures, it can be safely assumed that the LCDS is operating properly and that the cause of the malfunction is either in the application system hardware/software interface or in the circuits on the SC/MP CPU Application Card not associated with LCDS operation. If any indication cannot be obtained as specified for the Panel or TTY checkout procedures, return the LCDS to the factory for repair under the WARRANTY or SERVICE policy.

The Read/Write Memory locations specified in the program listing were chosen because they are not used by the LCDS DEBUG firmware. Thus, after the program is initially loaded via the Panel keyboard switches or the TTY, it will be retained in LCDS Read/Write Memory until power is turned off to the LCDS. If frequent use of the program is anticipated, it also may be desirable to punch the program onto paper tape as described in 3.4.7. Reloading of the program into LCDS Read/Write Memory then can be accomplished as described in 3.4.8.

Before the program may be called, the HALT INST switch on the LCDS must be set to DEBUG to enable normal termination of the program, and the following registers must be loaded with the specified values:

P1 - starting address of the data to be punched

P2 - high-order address of a software stack that allows at least 10 memory locations for use by the program; LCDS Read/Write Memory address X'77D0 may be used for this purpose

E - X'00 to select 256-location memory range or X'FF to select 512 location memory range

After the HALT INST switch is set to DEBUG and the P1, P2, and Extension Registers are loaded with the specified values, the program may be called by initiating LCDS RUN Mode operation at address X'7700 (refer to 3.3.1 and 3.4.2). The program then enters a wait state until the TTY tape punch is turned on and a (any) TTY key is pressed. Following these two actions, the program causes a complemented-binary paper tape to be punched out in the format shown previously. At the end of the trailer, the program automatically branches to the GETC routine of the LCDS DEBUG firmware so that the user may terminate program execution by turning off the TTY tape punch and pressing any TTY key. (When a TTY key is pressed after the tape punch is turned off, GETC returns to the Halt Instruction located at the end of the program and, since the HALT INST switch will be set to DEBUG, the LCDS will enter the DEBUG Mode via the hardware-forced DEBUG Save Routine; refer to 3.4.2.)

The length of the leader and trailer is determined by the program constant LEN contained in memory location 7704, and the complemented binary data is obtained via the instructions contained in memory locations 771B through 771E. All that is necessary to alter the length of the leader and trailer, therefore, is to load an appropriate new value into memory location 7704 (10_{10} = 1-inch leader/trailer). Similarly, if straight-binary rather than complemented-binary data is required, the instructions in memory locations 771B through 771F can be changed to the following:

771B	C101	LD	@1(P1)
771D	08	NOP	
771E	08	NOP	
771F	08	NOP	

National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, California 95051
(408) 737-5000
TWX: 910-339-9240

National Semiconductor GmbH
808 Fuerstenfeldbruck
Industriestrasse 10
West Germany
Telephone: (08141) 1371
Telex: 05-27649

NS Electronics (HK) Ltd.
4 Hing Yip Street, 11th Floor
Kwun Tong
Kowloon, Hong Kong
Telephone: 3-411241-8
Telex: 73866 NSE HK HX

NS International Inc.
Miyake Bldg. 6F, 1-9 Yotsuya
Shinjuku-Ku
Tokyo 160, Japan
Telephone: 03-355-3711
Telex: J28592

NS Electronics Pty. Ltd.
CNR-Stud Road & Mountain Highway
Bayswater, Victoria 3153, Australia
Telephone: 03-729-6333
Telex: 32096

Publication No. 4200105A